Analyse d'un algorithme de classification hiérarchique "en parallèle" pour le traitement de gros ensembles

Israël-César C. Lerman, Philippe Peter

IRIA

CENTRE DE RENNES
IRISA

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France
Tél. (3) 954 90 20

Rapports de Recherche

N° 339

ANALYSE D'UN ALGORITHME DE CLASSIFICATION HIÉRARCHIQUE "EN PARALLÈLE" POUR LE TRAITEMENT DE GROS ENSEMBLES

Israël-César LERMAN
Philippe PETER

Septembre 1984

IRISA — PUBLICATIONS INTERNES -

ANALYSE D'UN ALGORITHME DE CLASSIFICATION HIERARCHIQUE «EN PARALLELE» POUR LE TRAITEMENT DE GROS ENSEMBLES

Israël-César LERMAN
Philippe PETER

Publication n° 232 - Août 1984

IRISA

INSTITUT DE RECHERCHE EN INFORMATIQUE
ET SYSTÈMES ALÉATOIRES

Campus Universitaire de Beaulieu
Avenue du Général Leclerc
35042 - RENNES CÉDEX
FRANCE
Tél. : (99) 36.20.00
Télex : UNIRISA 95 0473 F

Publication Interne n° 232
Août 1984
114 pages

ANALYSE D'UN ALGORITHME DE CLASSIFICATION HIERARCHIQUE "EN PARALLELE" POUR LE TRAITEMENT DE GROS ENSEMBLES. ASPECTS METHODOLOGIQUES ET PROGRAMMATION.

Israël-César LERMAN et Philippe PETER

RESUME : Nous présentons dans cet article un algorithme approché de classification hiérarchique de gros ensembles offrant d'importantes possibilités de parallélisation. Cet algorithme peut s'appliquer à n'importe quelle méthode de base telle que par exemple l'A.V.L. (Algorithme de la Vraisemblance du Lien) ou à l'Inertie Expliquée.

Après avoir décrit notre nouvelle méthode, nous détaillons l'implémentation que nous en avons réalisée. Ce programme qui respecte les normes FORTRAN de MODULAD (document normalisation F. -66 version 1.1)- permet la classification d'individus décrits par des attributs de présence/absence. En outre, il est appliqué à l'A.V.L.

ABSTRACT : In this paper, we present a hierarchical classification approximate algorithm for large data sets. This algorithm offers important possibilities of "parallelization". It can be applied to any basic method as for example the explained variance or the likehood of the links.

After we described this new method, we detail the implementation we have realized. This program -according to FORTRAN MODULAD's standarts ("document normalization F.-66 version 1.1)- assumes the classification of objects described by attributes, it is applied to the likehood of the links algorithm.

CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE
(L.A. 227)
UNIVERSITÉ DE RENNES 1 I.N.S.A. DE RENNES

INSTITUT NATIONAL DE RECHERCHE
EN INFORMATIQUE ET EN AUTOMATIQUE
(LABORATOIRE DE RENNES)

PLAN

A - NOM ET OBJET :

Le programme CAHAP que nous présentons permet de faire une classification hiérarchique approchée d'individus décrits par des attributs.

Version 0.0 : JUILLET 1984

Auteurs : LERMAN I. César - Université de Rennes I,
PETER Philippe - Université de Rennes I.

Ce programme a bénéficié de nombreux sous-programmes mis au point par Henri LEREDDE (Université de Paris Nord) pour la version "MODULAD" du programme de classification automatique hiérarchique "CAHLR".

B - ASPECTS THEORIQUES :

UN ALGORITHME "PARALLELE" DE CLASSIFICATION HIERARCHIQUE DE "GROS ENSEMBLES".

I - IDEE GENERALE ET DIFFERENTES PHASES DE L'ALGORITHME.

L'information de base est définie par une table à double entrée qui représente la description d'un ensemble E d'objets -généralement spécifié par un échantillon de la population P étudiée- au moyen d'un ensemble V de variables. On suppose que l'ensemble des lignes (resp. colonnes) du tableau représente E (resp. V) ; de sorte qu'à l'intersection d'une ligne représentant l'objet x ($x \in E$) et d'une colonne associée à la variable v ($v \in V$), on trouve la valeur v(x) de v sur x.

Il y a plusieurs types de tableaux de données (cf. [LERMAN (1981)], Chap.2) et l'algorithme que nous allons présenter peut concerner le traitement de l'ensemble des lignes (resp. colonnes) de n'importe quel type de tables de données. Toutefois nous l'avons effectivement implémenté dans le cas d'un tableau d'incidence où V est formé de variables logiques de présence-absence : v(x)=1 (resp. 0) selon que l'objet possède (resp. ne possède pas) l'attribut défini par la variable v, $v \in V$.

Le cardinal de l'ensemble E des objets (card(E)) est le plus souvent plus important que celui de l'ensemble V des variables. Card(E) peut atteindre plusieurs milliers alors que card(V) dépasse rarement quelques centaines. De sorte que bien que la notion de "gros" ensemble soit toute relative et dépende de la configuration informatique dont on dispose, l'algorithme concerne surtout le problème de la classification de l'ensemble E des objets.

On a vu ces dernières années se développer des algorithmes rapides de classification ascendante hiérarchique basés sur une limitation qu'on démontre possible dans la comparaison des indices de proximité -les plus utilisés- entre paires de classes [propriété de "contractance" et graphes "réductibles" ([BRUYNOOGHE (1978)], [JAMBU (1978)]), détermination des "voisins réciproques" ([RAHM (1980)]), ...]. Ces algorithmes fournissent des résultats "exacts" ; c'est à dire, les mêmes que si l'algorithme classique -qui suppose la comparaison de toutes les paires de classes- est pratiqué.

Dans l'évaluation de ces algorithmes dits "rapides", on oublie trop souvent de tenir compte du temps nécessaire à l'établissement de la table des indices de proximité entre éléments de E ou de la préordonnance totale associée alors que pour l'algorithme classique appliqué dans son contexte de tailles "raisonnables" (quelques centaines sur de "grosses" machines), ce temps peut devenir très dominant si le nombre de variables (représentées ici par les colonnes) est "assez grand". Nous avons très clairement expérimenté ce fait en considérant le problème transposé de la classification hiérarchique de l'ensemble des variables où l'ensemble des objets pouvait atteindre quelques milliers.

L'algorithme que nous développons ici n'est pas exact, il fournit une forme d'"approximation" de l'arbre complet. Chaque feuille de

l'arbre construit est définie par une sous-classe homogène d'une grande classe correspondant à un même profil général qui "doit" nécessairement apparaître dans une classification hiérarchique exacte et totale. Ainsi, la structure principale en classes se retrouve -à partir de l'un des premiers niveaux- dans la structure de l'arbre que nous proposons. La qualité de l'approximation est de nature statistique et liée à la classifiabilité naturelle de l'ensemble décrit E.

La méthode que nous allons présenter de "Classification Automatique Hiérarchique Approchée en Parallèle (C.A.H.A.P.)" pourrait d'une certaine manière être considérée comme une illustration de la célèbre formule "diviser pour régner". Elle comporte quatre phases que nous reprenons en détail ci-dessous :

1- L'ensemble E des objets est découpé en sous-ensembles de tailles respectives comparables : E = E1 + E2 + .. + Ej + .. + Ek (somme ensembliste). Chaque sous-ensemble définira une tranche.

2- Chaque tranche fait l'objet d'une classification hiérarchique partielle ; c'est à dire, qu'on arrête à un niveau dépendant du nombre de classes formées et d'un critère d'adéquation. Une même classe formée à ce niveau de la tranche traitée définira ce que nous appellerons une "sous-classe".
   Il est important déjà de remarquer que toutes les tranches peuvent simultanément être organisées en "sous-classes" sur un calculateur parallèle.

3- La troisième phase consiste à classifier -par la même méthode que celle utilisée en 2- l'ensemble de toutes les sous-classes obtenues à partir du traitement des différentes tranches (phase 2 ci-dessus).

4- La quatrième et dernière phase correspond à une édition des résultats acquis dans les phases 2 et 3.

Cet algorithme composé peut a priori être utilisé avec n'importe quel critère de formation ascendante hiérarchique des classes. Toutefois, il importe que le critère n'ait pas une tendance naturelle à former des classes par trop inégales en taille. Un critère tel que celui de la vraisemblance du lien répond parfaitement au problème compte tenu d'un certain équilibre dans les cardinaux des classes qu'il permet de construire ([F.NICOLAU (1980)]).

L'idée de la réalisation de ce type d'algorithme remonte pour nous à l'année 1979 où, dans le cadre d'un stage de D.E.A. (C.N.E.T.-Lannion) ([RAPHALEN(1979)], [VALETTE & al. (1980)]) le problème s'était posé de typer la charge d'un ordinateur (il s'agissait de l'IRIS 80) et ce, à partir de la classification de plusieurs milliers d'"unités de travail", caractérisées par différents paramètres d'utilisation de la configuration informatique.

De façon tout à fait indépendante et dans un tout autre contexte Madame Escofier [ESCOFIER (1979)] a étudié et traité le problème de l'approximation d'une analyse factorielle des correspondances portant sur un "grand" tableau à partir d'analyses partielles de même type portant chacune sur un sous-tableau, où l'ensemble des sous-tableaux définit une partition du tableau complet. Toutefois les problèmes d'approximation posés dans le cadre factoriel sont de nature (technique ou statistique) très différente.

Comme nous l'avons signalé ci-dessus, nous allons reprendre et expliciter les quatre phases de notre algorithme.

II - DECOUPAGE EN TRANCHES.

Compte tenu des possibilités informatiques de traitement, l'utilisateur du programme fournit le nombre m d'objets par tranche. Supposons que I={1,2,...,i,...,n} indexe la suite des éléments. Conformément à la division n=m*(k-1)+l, la j-ième tranche est formée des objets d'indices [m*(k-1)+1] à m*j, j=1,2,...,(k-2). La dernière tranche, formée des objets d'indices [m*(k-1)+1] à n, comporte l termes et l peut être inférieur à m. Toutefois, en jouant sur le diviseur m, on s'arrangera pour -des raisons de précision statistique- que la dernière tranche soit d'une taille l "très comparable" à m.

Le choix des objets devant rentrer dans la composition d'une même tranche doit correspondre à un échantillon aléatoire exhaustif de la population P dont provient E. C'est automatiquement le cas si la suite (ei/1<=i<=n) des objets formant E est construite de telle sorte que ei (1<=i<=n) soit choisi indépendamment de (e1,e2,...,ei-1), uniformément au hasard dans (P-{e1,e2,...,ei-1}).

Relativement à cette décomposition aléatoire des problèmes intéressants se présentent concernant la stabilité statistique de la classification hiérarchique de l'ensemble des sous-classes.

III - CLASSIFICATION HIERARCHIQUE PARTIELLE D'UNE TRANCHE D=Ej.

L'algorithme de base choisi est celui de la Vraisemblance du Lien (A.V.L.) [LERMAN (1970) (1981), LERMAN - LEREDDE (1983)], dont nous rappellerons rapidement ci-dessous le critère (cf. paragraphe IV.1). Toutefois, dans cette phase on arrête le déroulement de l'algorithme à un certain niveau dont nous préciserons bientôt la règle de détermination. Nous n'avons en effet besoin que d'une partition en sous-classes.

III.1 - Règle d'arrêt.

Des paramètres fournis par l'utilisateur se déduit le nombre maximum total NCL de sous-classes à retenir à partir de la classification ascendante hiérarchique d'une même tranche. On dispose d'autre part d'un critère très général $C[\Pi, p(D)]$ mesurant l'adéquation entre une partition Π sur D et une structure de proximité p(D) -de caractère numérique ou ordinal- sur D. Dans ces conditions, la règle adoptée est la suivante :

On désigne par NCP1 (resp. NCP2) le premier niveau de l'arbre pour lequel le nombre de classes de la partition obtenue est inférieur ou égal à NCL [resp. (NCL-t)] où t est un entier "assez petit" devant NCL. Dans le programme implémenté où on a travaillé avec une valeur NCL de l'ordre de 120, on a pris t=5. On a bien entendu NCP1<NCP2. Le niveau de coupure NCP est alors choisi, compris entre NCP1 et NCP2 (NCP1<=NCP<=NCP2), de façon à optimiser le critère $C[\Pi, p(D)]$.

Cette technique permet de retenir la partition la plus "naturelle" dont le nombre de classes est inférieurement le plus voisin de NCL.

III.2 - Expression du critère de sélection de la partition.

Nous avons développé (voir par exemple dans [LERMAN (1981) et (1982)]) un critère très général d'adéquation entre une partition $\Pi(D)$ et une structure de proximité p(D), sur D. Le critère a un fondement clair aussi bien d'un point de vue formel que statistique. Trois situations différentes sont considérées en ce qui concerne la structure de proximité p(D) :

(a) p(D) est une "ordonnance" totale sur D (i.e. ordre total et strict sur l'ensemble P2(D) des paires d'éléments de D, où le rang d'une paire est une fonction strictement monotone de la ressemblance entre ses deux composantes).

(b) p(D) est une "préordonnance" totale sur D (i.e. préordre total sur l'ensemble P2(D) des paires d'éléments de D, où le rang d'une paire est une fonction strictement monotone de la

ressemblance entre ses deux composantes).

(c) p(D) est une "similarité" sur D, associant à chaque paire d'objets {x,y} (élément de P2(D)) un nombre Q(x,y) sensé "mesurer" la ressemblance entre x et y.

Généralement, le choix d'une structure de proximité de type (a) [resp. (b)] est associé au choix d'une structure de proximité de type (c). il y a certes un intérêt spécifique à utiliser une ordonnance ou une préordonnance (cf. [LERMAN (1981) Chap. 2,4 et 10]). Toutefois, compte tenu du soin que nous apportons dans la définition d'un indice de proximité, nous utiliserons directement la forme (c), ce qui permet d'éviter l'établissement de l'ordonnance au moyen d'un tri qui alourdirait (temps calcul et espace mémoire) l'algorithme général.

En adoptant la forme (c) de la structure de proximité, le critère $C(\Pi(D),p(D))$ peut être vu comme un cas particulier d'un indice très général d'association entre deux relations pondérées sur le même ensemble D [H.E. DANIELS (1944), L.J. HUBERT & F.B. BAKER (1978), G. LECALVE (1976), I.C. LERMAN (1976), N. MANTEL (1967)].

J étant l'ensemble -de cardinal m- qui indexe D, désignons par $\{q_{ij}/(i,j)\in JxJ\}$ (resp. $\{r_{ij}/(i,j)\in JxJ\}$) l'une (resp. l'autre) des pondérations où -pour des raisons de convenance technique- on pose $q_{ii}=0$ (resp. $r_{ii}=0$) pour tout i de J.

Le critère d'adéquation se présente sous la forme

$$[s - \mathcal{E}(S)]/\sqrt{\text{var}(S)} \qquad (1)$$

où

$$s = \sum_{1\leq i,j\leq m} q_{ij}\, r_{ij} \qquad (2)$$

et où S représente l'une des deux variables aléatoires (v.a.) duales et de même distribution

$$\sum_{1\leq i,j\leq m} q_{ij}\, r_{\sigma(i)\sigma(j)} \quad \text{et} \quad \sum_{1\leq i,j\leq m} q_{\sigma(i)\sigma(j)}\, r_{ij} \qquad (3)$$

dans l'expression desquelles σ est une permutattion aléatoire dans l'ensemble -muni d'une probabilité uniforme- des m! permutations sur (1,2,...,m).

On a :

$$\mathcal{E}(S) = \sqrt{A_1 B_1}/n(n-1) \qquad (4)$$

et

$$\mathrm{var}(S) = -A_1B_1/[n(n-1)]^2 + 2A_3B_3/n(n-1) + 4(A_2-A_3)(B_2-B_3)/n(n-1)(n-2) + (A_1-4A_2+2A_3)(B_1-4B_2+2B_3)/n(n-1)(n-2)(n-3) \qquad (5)$$

où :

$$A_1 = \Big(\sum_{1\leqslant i,j\leqslant m} q_{ij}\Big)^2, \quad A_2 = \sum_{1\leqslant i\leqslant m}\Big(\sum_{1\leqslant j\leqslant m} q_{ij}\Big)^2, \quad A_3 = \sum_{1\leqslant i,j\leqslant m} (q_{ij})^2$$

$$B_1 = \Big(\sum_{1\leqslant i,j\leqslant m} r_{ij}\Big)^2, \quad B_2 = \sum_{1\leqslant i\leqslant m}\Big(\sum_{1\leqslant j\leqslant m} r_{ij}\Big)^2, \quad B_3 = \sum_{1\leqslant i,j\leqslant m} (r_{ij})^2 \qquad (6)$$

On remarquera avec intérêt que le calcul du coefficient (1) nécessite simplement une double boucle DO sur $\{1,2,\ldots,m\}$.

Dans le cas qui nous intéresse ici de la comparaison d'une similarité à une partition, $\{rij/(i,j)\in JxJ\}$ correspond à la fonction indicatrice de la relation d'équivalence sur D définie par la partition $\Pi(D)=\{D1,D2,\ldots,Dh\}$. De sorte que rij=1 (resp. 0) si i et j appartiennent à une même classe Dg ($1\leqslant g\leqslant h$) (resp. sinon). Ainsi, désignant par mg le cardinal de la classe Dg, $1\leqslant g\leqslant h$, on a, parce que rii=0 pour tout i=1,2,...,m

$$B_1 = \Big[\sum_{1\leqslant g\leqslant h} mg(mg-1)\Big]^2, \quad B_2 = \sum_{1\leqslant g\leqslant h} mg(mg-1)^2 \text{ et } B_3 = \sum_{1\leqslant g\leqslant h} mg(mg-1) \qquad (7)$$

D'autre part, compte tenu de la symétrie de la relation de similarité, on a qij=qji pour tous $1\leqslant i,j\leqslant m$.

C'est l'expression (1) ci-dessus qui a été directement programmée. Toutefois, nous montrons dans [LERMAN (1981) Chap. 4 et (1983)] que le critère peut se mettre sous la forme :

$$\frac{1}{\sqrt{V(r')}} \sum \{r(u)\, c(u) \;/\; u \in J^{\{2\}}\} \qquad (8)$$

où nous désignons par $J^{\{2\}} = P2(J)$ l'ensemble des paires (i.e. des parties à deux éléments) de J, où r a la même signification que ci-dessus (fonction indicatrice de la relation partition) et où c est la mesure de similarité entrée et réduite globalement :

$$c(u) = [q(u)-\alpha]/\lambda \qquad (9)$$

où

$$\alpha = \frac{1}{M}\sum\{q(u)/u \in J^{\{2\}}\} \text{ et } \lambda^2 = \frac{1}{M}\sum\{[q(u)-\alpha]^2/u \in J^{\{2\}}\}$$

Il reste à préciser le sens de V(r´) : il s´agit de la variance de la variable aléatoire

$$\sum \{r'(u)\ c(u) \ /\ u \in J^{\{2\}}\} \qquad (10)$$

où r´ est la fonction indicatrice de la partition : élément aléatoire pris dans l´ensemble -muni d´une probabilité uniforme- P(m;t) des partitions de même type (suite des cardinaux des classes) t que la partition considérée. On obtient :

$$V(r') \cong \binom{(m)}{(2)}\Big(\sum_{1\leqslant g\leqslant h} \pi g^2\Big)\ [1 + \Big(\sum_{1\leqslant g\leqslant h} \pi g^2\Big)]$$

$$+[\ \sum \{c(u)\ c(v)/(u,v) \in G\}]*[\Big(\sum_{1\leqslant g\leqslant h} \pi g^3\Big)+\Big(\sum_{1\leqslant g\leqslant h} \pi g^2\Big)^2\ ] \qquad (11)$$

où $\pi g = (mg/m)$ est la proportion d´objets contenus dans la g-ième classe, $(1\leqslant g\leqslant h)$ et où G désigne l´ensemble des couples de paires d´objets de D, ayant une composante commune ; G = {({x,y},{x,z}) /x,y,z mutuellement distincts et appartenant à D}. G a pour cardinal m(m-1)(m-2).

En posant c(j,j)=0 pour tout j de J, on peut voir que la somme sur G -entre crochets ci-dessus peut se mettre sous la forme :

$$\sum \{c(i,j)\ c(i,g)\ /\ 1\leqslant i,j,g\leqslant m\} \qquad (12)$$

qu´on peut réduire à la forme suivante :

$$\sum_{1\leqslant i\leqslant m}\ [\ \sum_{1\leqslant j\leqslant m} c(i,j)]^2 - m(m-1) \qquad (13)$$

Ainsi, le calcul de la variance V(r´) dépend seulement d´une double boucle DO qui est indépendante de la partition $\pi(D)$ et qui peut être effectuée dès le départ après la détermination de la table $\{c(u) \ /\ u \in J^{\{2\}}\}$.

IV - ASSOCIATIONS ENTRE "SOUS-CLASSES".

IV.1 Chaque sous-classe est représentée par un noyau formé d'un seul élément.

C'est la solution la plus simple et qui a montré une très bonne efficacité dans le cadre de la méthode employée de la "Vraisemblance du Lien". Rappelons que l'algorithme de classification ascendante hiérarchique suppose la définition sur l'ensemble C à organiser d'un indice qui se réfère à une échelle [0,1] de probabilité (ou de fréquence mathématique) et dont la valeur reflète le complément à 1 d'un "degré d'invraisemblance" de la relation observée.

Si $\{P(x,y) / \{x,y\} \in P2(C)\}$ est la table des valeurs d'un tel indice d'association sur l'ensemble P2(C) des paires d'éléments distincts de C et si G et H représentent deux parties disjointes (i.e. deux classes) de C, l'indice de proximité entre G et H se met sous la forme :

$$(\max\{P(x,y)/\ (x,y) \in GxH\})^{\mathrm{Card}(G)*\mathrm{Card}(H)} \qquad (1)$$

Pour des raisons de précision de calcul et compte tenu du fait que seule l'échelle ordinale induite par (1) importe pour la formation de l'arbre, on considère la fonction strictement croissante $-\mathrm{Log}[-\mathrm{Log}(.)]$; ce qui donne pour le transformé de l'indice (1) :

$$-\mathrm{Log}[\mathrm{card}(G)]-\mathrm{Log}[\mathrm{card}(H)]-\mathrm{Log}[-\mathrm{Log}(\max\{P(x,y)/(x,y)\in GxH\})] \qquad (2)$$

Au paragraphe III ci-dessus, l'ensemble C se trouvait défini par une tranche D=Ej du "gros" ensemble E à classifier, alors qu'ici C est l'ensemble des représentants des différentes "sous-classes" obtenues par les classifications hiérarchiques partielles des tranches (cf. paragraphe III).

IV.1.1 - Choix du "meilleur" représentant d'une sous-classe.

Compte tenu du fait que dans "A.V.L." nous travaillons avec un indice de proximité entre éléments (de l'ensemble à classifier) ayant une forme corrélative (ayant la référence à une échelle de probabilité) et en se souvenant que la nature du critère maximisé en Analyse en Composantes Principales est une somme des carrés de corrélations, nous choisirons comme "meilleur" représentant d'une sous-classe, l'élément dont la somme des carrés des indices d'association avec les autres éléments de la sous-classe, est maximale.

Rappelons rapidement que dans la situation implémentée de la classification d'un ensemble d'objets décrits par des attributs descriptifs, l'indice de proximité utilisé entre deux objets x et

y, se met sous la forme :

$$Q(x,y) = \{s(x,y)-[p(x)p(y)/p]\}/\sqrt{[p(x)p(y)/p]} \qquad (3)$$

où p est le nombre total d'attributs, p(x) [resp. p(y)] est le nombre d'attributs possédés par l'objets x (resp. y) et s(x,y) est le nombre d'attributs possédés en commun par les objets x et y.

Dans l'algorithme implémenté chaque représentant est affecté du poids-cardinal unité. Mais rien n'empêche à ce qu'il puisse être affecté du poids cardinal de la sous-classe qu'il représente. Dans ce cas, la table des indices d'association entre représentants (de sous-classes) fournie à l'A.V.L. sera accompagnée de la table des cardinaux des sous-classes et sera -après transformation conformément à la formule (2) ci-dessus- considérée comme la table des indices d'association entre sous-classes.

Plus précisément, si les indices d'association entre représentants sont donnés -dans le cadre d'une table- sous la forme (3), on commencera par une "réduction globale des similarités" où on substitue à la table

$$\{ Q(x,y) \ / \ \{x,y\} \in P2(C)\} \qquad (4)$$

celle

$$\{Q(x,y) = \frac{Q(x,y)-\bar{Q}}{\sqrt{var(Q)}} \ / \ \{x,y\} \in P2(C)\} \qquad (5)$$

où $\bar{Q}$ et var(Q) sont la moyenne et la variance de la distribution observée (4).

à partir de (5), on obtient :

$$\{P(x,y) = \Phi [Qr(x,y)] \ / \ \{x,y\} \in P2(C)\} \qquad (6)$$

où Φ est la fonction de répartition de la loi normale centrée et réduite.

Soient D1 et D1' deux sous-classes quelconques représentées par x et x'. Si on veut tenir compte des cardinaux des sous-classes, dans leurs comparaisons mutuelles au moyen de leurs représentants respectifs, on remplacera la dernière partie de l'expression (2) par {-Log[-Log(P(x,x')]}, et la première partie par {-Log[card(D1)]-Log[card(D1')]}.

Encore une fois, pour l'algorithme implémenté la suite se passe comme si card(D1) et card(D1') sont égaux à 1.

IV.1.2 - Usage du critère de l'inertie expliquée.

Nous avons déjà signalé que -à la différence de l'algorithme du lien unique ("single linckage")- l'A.V.L. ne subissait pas le phénomène bien connu du "chaînage" et avait une tendance naturelle à fournir des classes de tailles respectivement équilibrées. Toutefois, on peut parfaitement envisager le même processus avec d'autres critères. Le plus connu est sans doute celui de l'inertie expliquée [WARD (1963)] qui suppose une représentation euclidienne de l'ensemble des unités de données.

p désignant toujours le nombre d'attributs, on supposera $\{0,1\}^p$ muni d'une métrique $\{mj/1 \leqslant j \leqslant p\}$ diagonale (e.g. $mj=1/\varrho j$ où ϱj est la proportion d'objets possédant le j-ème attribut). La distance entre les deux objets x et y s'écrit dans ces conditions

$$d^2(x,y) = \sum_{1 \leqslant j \leqslant p} mj(xj-yj)^2 \qquad (7)$$

où xj (resp. yj) vaut 1 ou 0 selon que l'objet x possède ou non le j-ème attribut.

Le critère d'association entre deux classes G et H se met sous la forme

$$\frac{card(G)*card(H)}{card(G)+card(H)}\, d^2[g(G),g(H)] \qquad (8)$$

où g(G) [resp. g(H)] est le centre de gravité de la classe G (resp. H).

Enfin, le représentant d'une "sous-classe" sera déterminé comme minimisant la somme des carrés des distances aux autres éléments de la sous-classe.

IV.2 - Chaque sous-classe est prise globalement.

C'est une solution qui peut être envisagée avec l'un ou l'autre des deux critères principaux ci-dessus présentés. La table des indices d'association entre "sous-classes" d'une même tranche est déjà établie dans les classifications hiérarchiques partielles. Il reste à organiser et à établir la table des indices d'association entre "sous-classes" de tranches distinctes. Pour cela, on aura besoin à un moment donné de garder en "mémoire centrale" la table de croisement entre deux tranches organisées chacune en "sous-classes". le croisement d'une sous-classe G de l'une des tranches et d'une sous-classe H de l'autre tranche permet de déterminer la valeur de la quantité critère (2) ou (8), selon la méthode utilisée.

V - PRESENTATION GENERALE DES RESULTATS.

V.1 - Niveaux significatifs de l'arbre des classifications.

Nous avons l'habitude dans une analyse "fine" de distinguer deux notions (cf. par exemple [LERMAN (1981) ou (1983)]). La première concerne les "niveaux significatifs" et la seconde, les "noeuds significatifs". Les niveaux significatifs correspondent aux maxima locaux de la distribution observée sur la suite des niveaux d'une statistique "globale" telle que celle présentée dans le paragraphe III.2. Alors que les noeuds significatifs correspondent aux maxima locaux du taux d'accroissement de la statistique globale entre un niveau et le suivant.

Dans la présentation qu'on propose ici seuls les niveaux significatifs sont retenus.

V.2 - Présentation de l'arbre "global" des classifications.

Dans cet arbre général, chaque feuille se trouve définie par un sous-arbre obtenu à la phase deux (cf. paragraphe I et III). Ce sous-arbre se trouve représenté sur un seul niveau, mais où se trouvent marqués les numéros des niveaux des agrégations successives (cf. dans la figure ci-dessous le premier niveau du dernier arbre représenté).

Au delà de ce premier niveau qui reprend toutes les sous-classes "organisées", il est nécessaire d'augmenter artificiellement d'une constante, les niveaux de l'Arbre de Classification des Sous-Classes (ACSC) (i.e. des représentants), pour éviter toute confusion entre les niveaux des sous-arbres et ceux de l'ACSC. Cette constante doit être un majorant du plus haut niveau atteint par un sous-arbre. On peut carrément prendre le cardinal d'une tranche complète (dans l'exemple illustratif de la figure ce cardinal est égal à 5). Finalement pour le dessin de l'arbre global les niveaux retenus sont :

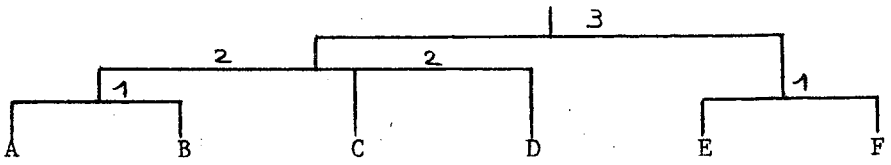
- le plus haut niveau de l'arbre global,
- les niveaux significatifs de l'ACSC ; mais où se trouvent marqués latéralement les agrégations entre deux niveaux significatifs,
- un niveau fictif correspondant à un majorant du plus haut niveau atteint par un sous-arbre définissant une sous-classe, de manière à ce que les classes formées au premier niveau sur le dessin correspondent aux sous-classes.

Les autres niveaux sont réhaussés -pour le dessin de l'arbre- au niveau immédiatement supérieur retenu.

Précisons que l'édition de l'arbre s'effectue à partir de sa représentation polonaise où on distingue des entiers négatifs correspondant à des niveaux d'agrégation entre deux classes et des entiers positifs correspondant aux codes des feuilles.

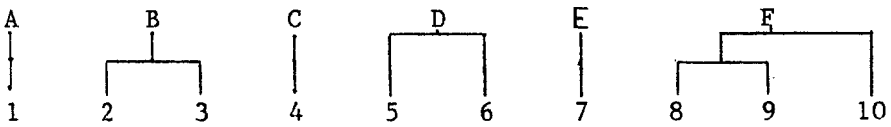
Soit l'arbre de classification des sous-classes :

-3 -2 -2 -1 A B C D -1 E F où A,..,F sont des sous-classes

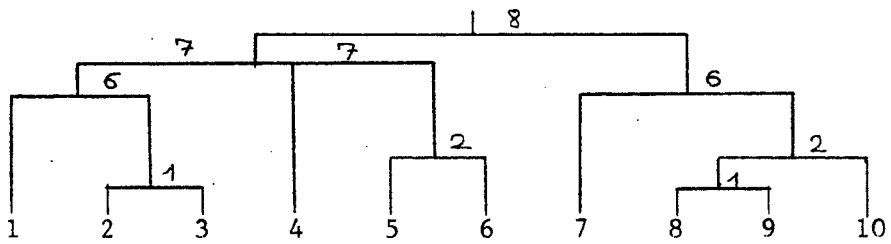


Sous-classes :

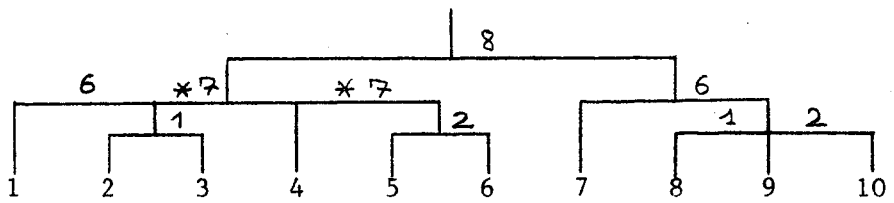
A: 1 ; B: -1 2 3 ; C: 4 ; D: -2 5 6 ; E: 7 ; F: -2 -1 8 9 10



Arbre détaillé (soit 5 le cardinal d'une tranche complète)



Arbre global dessiné (en supposant le niveau 7 -ancien niveau 2- significatif) :



VI - CONCLUSION.

La mise au point du programme a été le plus directement motivée par l'organisation d'un "gros" corpus de "petites annonces" concernant le marché de l'immobilier. Pour ce problème il était important d'avoir des sous-classes de taille "raisonnable" qu'il s'agira ensuite de proposer à l'utilisateur d'un système de consultation automatique. L'organisation par proximité de l'ensemble des sous-classes conformément à un arbre hiérarchique permet une recherche efficace de la "meilleure" sous-classe et le passage d'une sous-classe à celle voisine.

Il doit exister d'autres types d'applications -en reconnaissance d'images par exemple- où on souhaite d'un algorithme de classification qu'il puisse rapidement produire plusieurs "ilôts" denses qui s'associeraient naturellement par la suite ; alors que les algorithmes connus vont directement vers la recherche de la structure de proximité en les quelques grandes classes.

Ces derniers sont "directs", alors que notre algorithme fournit une approche des "grandes classes" en deux temps et nous avons mentionné tout l'intérêt de l'étude statistique de la stabilité des résultats. De toute façon, les résultats expérimentaux sont excellents.

Enfin et surtout, notre algorithme est susceptible d'une implémentation sur calculateur parallèle, ce qui permet d'en faire un algorithme très rapide et de faible encombrement mémoire.

BIBLIOGRAPHIE

M. BRUYNOOGH (1978) "Classification ascendante hiérarchique de grands ensembles de données : un algorithme rapide fondé sur la construction des voisinages réductibles", Cahiers de l'Analyse des Données, vol. III, numéro 1, 7-33.

H.E. DANIELS (1944) "The relation between measures of correlation in the universe of sample permutations", Biometrika, vol. 33, (1944).

B. ESCOFIER (1979) "Stabilité et approximation en analyse factorielle", Thèse d'état, 8.10.1979, Université Paris 6.

L.J. HUBERT & F.B. BAKER (1978) "Evaluating the conformity of sociometric measurements", Psychometrika, 43, 1, 31-41.

M. JAMBU (1978) "Classification automatique pour l'analyse des données", tome 1, Dunod, Paris.

G. LECALVE (1976) "Problèmes d'analyse des données", Thèse d'état 2ème partie, Université de Rennes I.

H. LEREDDE & I.C. LERMAN (1983) "CAHM1 : Méthode de classification hiérarchique", MODULAD, Bibliothèque Fortran pour l'Analyse des données, Version 1.0.

I.C. LERMAN (1970) "Sur l'analyse des données préalable à une classification automatique. Proposition d'une nouvelle mesure de similarité", Revue Mathématique et Sciences Humaines, 8ème année, numéro 32.

I.C. LERMAN (1976) "Formal analysis of a general notion of proximity between variables" in Proceed. (published by North Holland in 1977) of "Congrès Européen des Statisticiens", Grenoble.

I.C. LERMAN (1981) "Classification et analyse ordinale des données", Dunod, Paris.

I.C. LERMAN (1983) "Sur la signification des classes issues d'une classification automatique", NATO ASI Series, vol. G1, Numerical Taxonomy. Edited by J. Felsenstein, Springer-Verlag.

M. MANTEL (1967) "The detection of disease clustering and a generalised regression approach", Cancer research, 27, 209-220.

F. NICOLAU (1980) "Critérios de análise classificatória hierárquica baseados na função de distribuição", Laboratoire de Statistique, Faculté des Sciences de Lisbonne.

C. de RAHM (1980) "La classification hiérarchique ascendante selon la méthode des voisins réciproques", Cahiers de l'Analyse des Données, vol. V, numéro 2, 135-144.

M. RAPHALEN (1979) "Caractérisation de la charge du calculateur IRIS 80 du C.N.E.T. LANNION. Classification des travaux soumis." Rapport de D.E.A., Université de Rennes I.

N. VALETTE, A. DUPUIS & A. LELIEVRE (1980) "Application des techniques d'analyse des données à la caractérisation de la charge de l'IRIS 80.", Rapport C.N.E.T. RP/LAA/STI/15.

J.H., Jr. WARD (1963) "Hierarchical grouping to optimise an objective function", JASA, 58, 236-244.

C - PARAMETRES DE FONCTIONNEMENT :

C1) Cartes (où lignes) à founir :

Dans le cas où un programme conversationnel ne se charge pas de demander à l'utilisateur les paramètres nécessaires, il faut les introduire de la manière suivante sur le fichier numéro 41 :

Première ligne : titre de la classification (sur 80 caractères au maximum).

Deuxième ligne : PARAM(I), I=1,...,10 (format 10I5).

Troisième ligne : format des données (sur 80 caractères au maximum).

LES LIGNES SUIVANTES NE SONT NECESSAIRES QUE SI L'ON FOURNIT LES NOMS DES INDIVIDUS

Quatrième ligne : format de lecture des noms des individus (sur 80 caractères au maximum).

N lignes suivantes : un nom d'individu par ligne (N est le nombre d'individus).

Dernière ligne : format d'écriture d'une ligne de l'arbre de la forme : (n1x,n2a4,2x,105a1) avec :
n2 = ICAR (voir PARAM (10)) et
n1 = 25-(4*n2)

C2) Description du vecteur de paramètres :

PARAM(1) : nombre total d'individus.

PARAM(2) : nombre d'attributs.

PARAM(3) : hypothèse d'absence de lien choisie :

---> 1 : modèle poissonien (h.a.l.p.),

---> 2 : modèle hypergéométrique (h.a.l.h.).

PARAM(4) : réduction choisie :

---> 1 : réduction statistique globale des similarités (centrage et réduction).

---> 2 : réduction par homothétie de rapport lambda.

PARAM(5) : nombre d'individus par tranche complète.

PARAM(6) : sauvegarde de la matrice des similarités :

---> 0 : pas de sauvegarde,

---> 61 : sauvegarde sur le fichier numéro 61.

PARAM(7) : nombre total de classes permises.

PARAM(8) : option de sortie :

---> 1 : représentation polonaise de l'arbre de classification des classes, niveaux significatifs de cet arbre et composition des classes.

---> 2 : idem option 1 avec en plus le dessin de l'arbre de la classification des classes.

---> 3 : idem option 1 avec la représentation polonaise de l'arbre "global" et le dessin de cet arbre.

PARAM(9) : noms des individus :

---> 0 : l'utilisateur les fournit (possible uniquement quand PARAM(8)=3).

---> 1 : l'utilisateur ne les fournit pas : soit ils sont inutiles, soit ils sont générés par le programme (ELEMENT 1, ELEMENT 2, ELEMENT 3, ..., ELEMENT N).

PARAM (10) : nombre de mots mémoire nécessaires pour contenir le nom d'un individu :

---> 0 : l'utilisateur ne fournit pas les noms des individus.

---> ICAR : l'utilisateur fournit les noms des individus, ICAR = [(LONG+3)/4] où LONG est le nombre de caractères du plus long nom.

D - ASPECTS INFORMATIQUES :

D1) Gestion de mémoire :

Elle est faite par simulation d'allocation dynamique en "piochant" dans des super-tableaux (INTG pour les entiers, REEL pour les réels, LOGI pour les booléens et ICHN pour les caractères). Ceci est assuré par le sous-programme ALLOC.

Deux réallocations sont faites au moment de la présentation des résultats :

La première pour la présentation des résultats "texte" de manière à éliminer les tableaux qui ne sont plus nécessaires et à dimensionner les autres avec leur dimension exacte et non plus un majorant (comme par exemple pour le tableau des niveaux significatifs). Elle est à la charge du sous-programme TASSO.

La seconde (uniquement lorsque le dessin d'un arbre a été demandé) élimine les tableaux devenus inutiles et assure la simulation de l'allocation dynamique pour les tableaux nécessaires au dessin de l'arbre. Elle est à la la charge du sous-programme TASS1 s'il faut dessiner l'arbre de classification des classes ou du sous-programme TASS2 s'il faut dessiner l'arbre global.

D2) Indications de performances :

13 minutes CPU sous HB68/MULTICS pour classifier 1000 individus codés chacun avec 86 attributs (trois tranches d'environ 330 individus).

E) SOUS-PROGRAMMES REQUIS :

Sous-programme principal :
CLMIL.

Sous-sous-programmes principaux :
CLAS1 - CLAS2.

Sous-programmes d'initialisations :
LIPAR - LINOM - INIT.

Sous-programmes d'allocation dynamique :
ALLOC - TASS0 - TASS1 - TASS2.

Sous-programmes de lecture des données :
LCDON - LCDNG.

Sous-programmes de gestion des classifications :
CLS11 - CLS22.

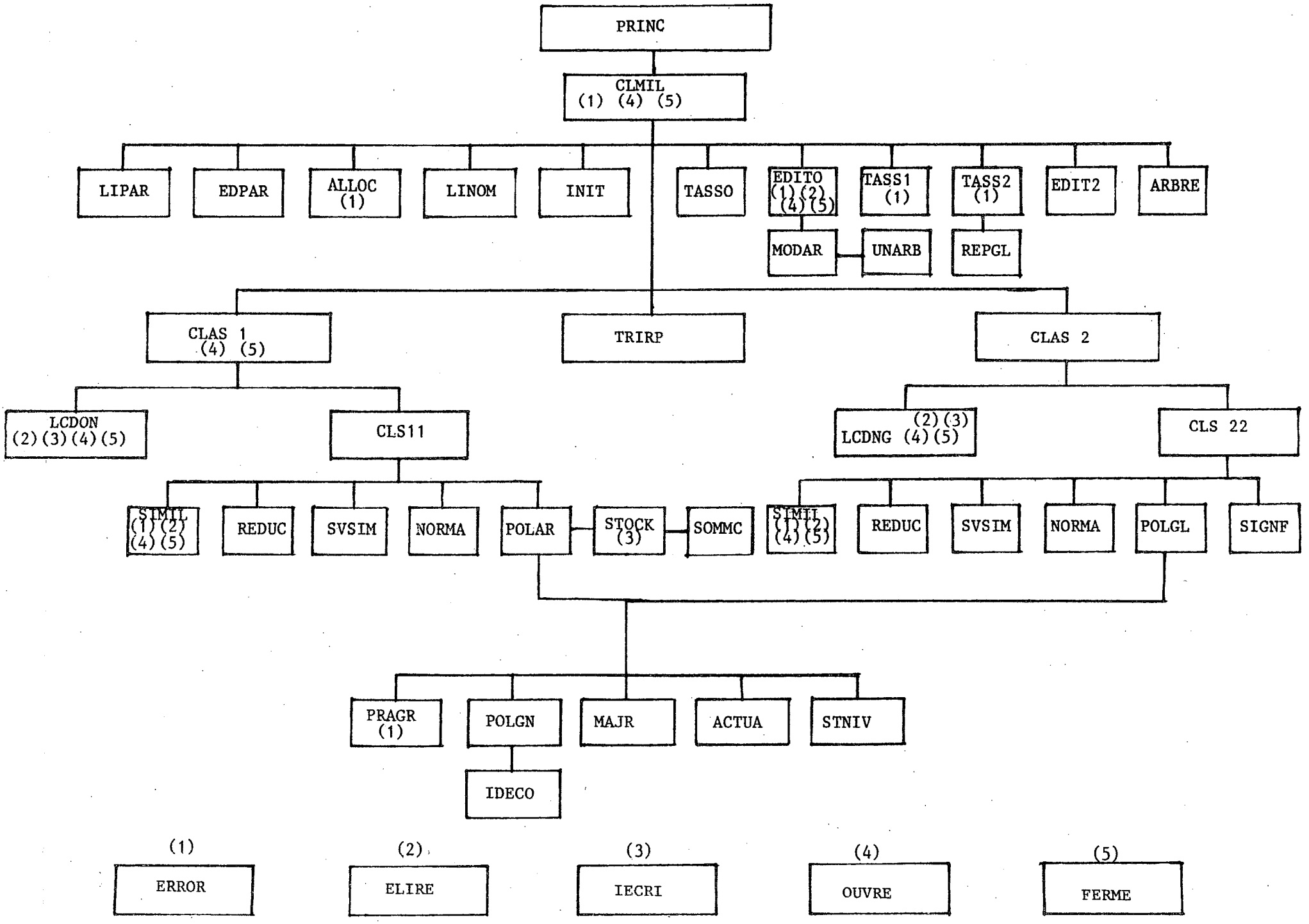
Sous-programmes de calcul pour la classification :
SIMIL - REDUC - NORMA - POLAR - POLGL - PRAGR - POLON - MAJR - ACTUA - STNIV - SIGNF - STOCK - SOMMC.

Sous-programmes de présentation des résultats :
MODAR - UNARB - REPGL.

Sous-programmes d'édition des paramètres ou des résultats :
EDPAR - SVSIM - EDIT0 - EDIT2 - ARBRE.

Sous-programme intermédiaire entre les classifications partielles et la classication globale :
TRIRP.

Sous-programmes utilitaires :
IDECD - ERROR - ILIRE - IECRI - OUVRE - FERME.


PRINC
CLMIL (1) (4) (5)
LIPAR
EDPAR
ALLOC (1)
LINOM
INIT
TASSO
EDITO (1) (2) (4) (5)
TASS1 (1)
TASS2 (1)
EDIT2
ARBRE
MODAR
UNARB
REPGL
CLAS 1 (4) (5)
TRIRP
CLAS 2
LCDON (2) (3) (4) (5)
CLS11
LCDNG (2) (3) (4) (5)
CLS 22
SIMIL (1) (2) (4) (5)
REDUC
SVSIM
NORMA
POLAR
STOCK (3)
SOMMC
SIMIL (1) (2) (4) (5)
REDUC
SVSIM
NORMA
POLGL
SIGNF
PRAGR (1)
POLGN
MAJR
ACTUA
STNIV
IDECO
(1)
ERROR
(2)
ELIRE
(3)
IECRI
(4)
OUVRE
(5)
FERME

F - FICHIERS UTILISES :

Le sous-programme principal n'utilise pas directement de fichiers.

fichier des paramètres (41) :
utilisé par les sous-programmes LIPAR et LINOM ; fichier d'entrée avec format.

fichier des données (11) :
utilisé par les sous-programmes LCDON et LCDNG ; fichier d'entrée avec format.

fichier des résultats "texte" (42) :
utilisé par les sous-programmes EDPAR, ERROR, EDITO et EDIT2 ; fichier de sortie avec format.

fichier de sauvegarde des tables de similarités (61) :
utilisé par le sous-programme SVSIM ; fichier de sortie avec format.

fichiers d'édition du dessin de l'arbre (46 et éventuellement 47, 48 et 49) :
utilisés par le sous-programme ARBRE ; fichiers de sortie avec format.

fichier d'une tranche des données (43) :
utilisé par les sous-programmes LCDON et LCDNG ; fichier de travail sans format.

fichier transposé d'une tranche des données (44) :
utilisé par les sous-programmes LCDON, LCDNG et SIMIL ; fichier de travail sans format.

fichier de stockage des représentations polonaises des classifications partielles des tranches (45) :
utilisé par les sous-programmes STOCK et EDITO ; fichier de travail sans format.

En outre tous ces fichiers sont traités par les sous-programmes OUVRE et FERME, les fichiers de travail sans format (43,44 et 45) étant accédés par les sous-programmes spécialisés ILIRE et IECRI.

G - TRANSMISSION ET VALIDATION DES DONNEES :

Les données doivent être placées dans le fichier 11. Une ligne de ce fichier doit correspondre au codage d'un individu par des attributs (format de lecture dans le fichier des paramètres).

Echanges d'information sur supports magnétiques :

Sur fichier 43 : recopie des données de la tranche courante ; écriture et exploitation par les sous-programmes de lecture des donnés LCDON et LCDNG.

Sur fichier 44 : fichier transposé des données de la tranche courante ; écrit par les sous-programmes de lecture des données LCDON et LCDNG puis exploité par le sous-programme de calcul des indices de similarité SIMIL.

Sur fichier 45 : sauvegarde des représentations polonaises des classifications partielles des tranches ; écriture faite par le sous-programme STOCK et exploitation par le sous-programme d'édition des résultats EDITO.

H - GESTION DES ERREURS :

Chaque détection d'erreur entraîne la modification d'une variable indicatrice d'erreurs et l'appel au sous-programme ERROR. Ce sous-programme écrit un message sur le fichier de sortie "texte" et rend la main au sous-programme appelant. La détection d'une erreur entraîne généralement l'abandon du programme.

Principales erreurs détectées :

Dimension de tableau insuffisante : impression de la dimension nécessaire.

Individus ne possédant aucun attribut (entraînerait une division par zéro).

ASPECTS INFORMATIQUES :

DOSSIER DE PROGRAMMATION

1) Sous-programme CAHAP :

2) Objet : dimensionnement des super-tableaux, appel au sous-programme principal CLMIL.

3) Description des arguments :

aucun.

4) Sous-programme(s) requis :

le sous-programme principal CLMIL.

5) Sous-programme(s) appelant(s) :

aucun.

6) Transmission des données :

ni espaces communs, ni équivalences, ni fichiers directement utilisés.

7) Divers :

néant.

1) Sous-programme CLMIL :

2) Objet : c'est en fait le sous-programme principal. Il gère les sous-programmes de lecture et d'édition des paramètres, d'initialisation, des classifications partielles et globale ainsi que ceux d'édition des résultats. Il assure en outre certaines initialisations comme les numéros des fichiers utilisés et celles des variables directement déduites des paramètres.

3) Description des arguments :

3-1) . LGINTG INTEGER (E) : dimension du super-tableau contenant les entiers.

3-2) . LGREEL INTEGER (E) : dimension du super-tableau contenant les réels.

3-3) . LGLOGI INTEGER (E) : dimension du super-tableau contenant les booléens.

3-4) . LGICHN INTEGER (E) : dimension du super-tableau contenant les chaînes de caractères.

3-5) . INTG INTEGER (T) : super-tableau contenant les entiers.

3-6) . REEL REAL (T) : super-tableau contenant les réels.

3-7) . LOGI LOGICAL (T) : super-tableau contenant les booléens.

3-8) . ICHN INTEGER (T) : super-tableau contenant les chaînes de caractères.

3-9) . NUMERR INTEGER (ER/ES) : code des erreurs.

4) Sous-programme(s) requis :

LIPAR - EDPAR - ALLOC - LINOM - INIT - CLAS1 - TRIRP - CLAS2 - TASSO - EDITO - TASS1 - TASS2 - EDIT2 - ARBRE - OUVRE - FERME - ERROR.

5) Sous-programme(s) appelant(s) :

Il est uniquement appelé par le programme principal CAHAP.

6) Transmission des données :

- ni espaces communs, ni équivalences ;

- fichiers utilisés :

Ce sous-programme n'utilise pas directement de fichiers, mais initialise leurs numéros pour toute la suite du programme. Ils sont rangés dans le tableau IFICH de la manière suivante :

IFICH (1) : 41 fichier des paramètres, repéré également par la variable IENT.

IFICH (2) : 42 fichier de sortie "texte", repéré également par la variable ISORT.

IFICH (3) : 11 fichier des données.

IFICH (4) : 43 fichier temporaire binaire destiné à contenir le codage de la tranche courante des individus.

IFICH (5) : 44 fichier temporaire binaire destiné à contenir la transposition du fichier précédant.

IFICH (6) : 45 fichier temporaire binaire destiné à contenir les représentations polonaises partielles.

IFICH (7) : 46 fichier de sortie du dessin de l'arbre.

IFICH (8) : 47 éventuellement suite du dessin de l'arbre.

IFICH (9) : 48 éventuellement suite du dessin de l'arbre.

IFICH (10) : 49 éventuellement suite du dessin de l'arbre.

7) Divers :

- Principales variables définies pour le reste du programme :

* tableau des paramètres :

IPARAM (1) : (ou NIND) nombre d'individus,

IPARAM (2) : (ou NATT) nombre d'attributs,

IPARAM (3) : (ou IHAL (1)) hypothèse d'absence de lien choisie,

IPARAM (4) : (ou IREDUC (1)) réduction globale choisie,

IPARAM (5) : (ou ICARDT) cardinal d'une tranche complète,

IPARAM (6) : (ou IFSIM (1)) =61 : sauvegarde de la matrice des similarités sur le fichier 61, sinon =0 (pas de sauvegarde),

IPARAM (7) : (ou NTOTAL) nombre total de classes permises,

IPARAM (8) : (ou IOPSOR) option de sortie,

IPARAM (9) : (ou INOM) détermine si l'utilisateur fournit le nom des individus (=0 dans ce cas, 1 sinon),

IPARAM (10) : (ou ICAR) nombre maximum de mots mémoire pour stocker le nom d'un individu,

IPARAM (11) : (2) nombre de tranches complètes,

IPARAM (12) : (2) cardinal de la tranche incomplète.

(1) variable utilisée dans la suite du programme mais pas dans ce sous-programme,

(2) variable calculée et non pas lue par un sous-programme.

* MXSYM = max(ICARDT,NTOTAL) : dimension de tableau,

* MXPAIR nombre maximum de paires de classes agrégeables au même niveau de l'arbre (dimension de tableau),

* NIPOL =2*MXSYM dimension du tableau de la représentation polonaise de l'arbre de classification,

* MIQ dimension du tableau de la matrice triangulaire inférieure des similarités,

* MIQ2 dimension du tableau de la matrice triangulaire inférieure stricte des similarités.

1) Sous-programme ALLOC :

2) Objet : sous-programme d'allocation dynamique. Ce sous-programme fournit dans le super-tableau du type correspondant l'adresse du premier élément de tous les tableaux utilisés dans la suite du programme (jusqu'à l'édition des résultats où une autre allocation dynamique permettra de gagner de la place). Ce sous-programme vérifie en outre que la taille des super-tableaux est suffisante, dans le cas contraire, il s'agit d'une erreur fatale.

3) Description des arguments :

3-1) . ISORT INTEGER (E) : numéro du fichier de sortie "texte" en cas d'erreur.

3-2) . LGINTG INTEGER (E) : dimension du super-tableau contenant les entiers.

3-3) . LGREEL INTEGER (E) : dimension du super-tableau contenant les réels.

3-4) . LGLOGI INTEGER (E) : dimension du super-tableau contenant les booléens.

3-5) . LGICHN INTEGER (E) : dimension du super-tableau contenant les caractères.

3-6) . NIND INTEGER (E) : nombre total d'individus.

3-7) . NATT INTEGER (E) : nombre d'attributs.

3-8) . ICARDT INTEGER (E) : cardinal d'une tranche complète d'individus.

3-9) . MXSYM INTEGER (E) : nombre maximum d'individus à classifier ensembles.

3-10) . NTOTAL INTEGER (E) : nombre maximum de classes.

3-11) . MXPAIR INTEGER (E) : nombre maximum de paires de classes pouvant être agrégées au même niveau de l'arbre.

3-12) . INOM INTEGER (E) : indique si l'utilisateur fournit le nom des individus.

3-13) . ICAR INTEGER (E) : nombre maximum de mots mémoire nécessaires pour stocker le nom d'un individu.

3-14) . MIQ INTEGER (E) : dimension du tableau de la matrice triangulaire inférieure des similarités.

3-15) . MIQ2 INTEGER (E) : dimension du tableau de la matrice triangulaire inférieure stricte des similarités.

3-16) . IQ INTEGER (S) : début du tableau de la matrice triangulaire inférieure des similarités.

3-17) . IQ2 INTEGER (S) : début du tableau de la matrice triangulaire inférieure stricte des similarités.

3-18) . IXMU INTEGER (S) : début du tableau de la contribution des individus à la moyenne pour l'hypothèse d'absence de lien entre individus.

3-19) . ISMU INTEGER (S) : début du tableau de la contribution des individus à la variance pour l'hypothèse d'absence de lien entre individus.

3-20) . ICARRE INTEGER (S) : début du tableau de la somme des carrés des similarités pour chaque individu.

3-21) . IXNIV INTEGER (S) : début du tableau de l'indice des niveaux de l'arbre.

3-22) . ILSYM INTEGER (S) : début du tableau facilitant l'adressage dans les tables de similarités.

3-23) . ILECT INTEGER (S) : début du tableau pour la lecture d'une ligne du fichier transposé des données.

3-24) . ILADR INTEGER (S) : début du tableau facilitant le repérage d'une classe lors de la construction de la représentation polonaise.

3-25) . INVSG INTEGER (S) : début du tableau des niveaux significatifs.

3-26) . IIDBZ INTEGER (S) : début du tableau du début des classes pendant la construction de la représentation polonaise.

3-27) . IIFNZ INTEGER (S) : début du tableau de la fin des classes pendant la construction de la représentation polonaise.

3-28) . IICDZ INTEGER (S) : début du tableau du cardinal des classes pendant la construction de la représentation polonaise.

3-29) . IIPOL INTEGER (S) : début du tableau de la représentation polonaise.

3-30) . IIPOL2 INTEGER (S) : début d'un tableau de travail pour la construction de la représentation polonaise.

3-31) . IIPAI1 INTEGER (S) : début du tableau qui pour chaque paire de classes agrégeables à un niveau contient la première classe.

3-32) . IIPAI2 INTEGER (S) : début du tableau qui pour chaque paire de classes agrégeables à un niveau contient la deuxième classe.

3-33) . IX1 INTEGER (S) : début du tableau contenant le codage d'un individu lors de la transposition des données.

3-34) . IY1 INTEGER (S) : début du tableau contenant une colonne du fichier des données lors de la transposition des données.

3-35) . IQ1 INTEGER (S) : début d'un tableau de travail pour la transposition des données.

3-36) . MIQ1 INTEGER (S) : dimension du tableau de travail pour la transposition des données.

3-37) . IREP INTEGER (S) : début du tableau des représentants des classes.

3-38) . IOREP INTEGER (S) : début du tableau assurant la correspondance entre les classes et leurs représentants après le tri de ces derniers.

3-39) . IICHMP INTEGER (S) : début d'un tableau de travail pour le tri des représentants.

3-40) . IITEM INTEGER (S) : début du tableau des noms des individus.

3-41) . LPREST INTEGER (S) : début du tableau de booléens pour le tri (par ventilation) des représentants.

3-42) . ICHFIN INTEGER (S) : premier emplacement libre dans le tableau des caractères.

3-43) . NUMERR INTEGER (ER/ES) : code des erreurs.

4) Sous-programme(s) requis :

ERROR.

5) Sous-programme(s) appelant(s) :

appelé par le sous-programme principal CLMIL.

6) Transmission des données :

ni espaces communs, ni équivalences, ni fichiers directement utilisés.

7) Divers :

Les tableaux servant pendant tout le programme (classifications partielles et globale) sont placés en tête des super-tableaux.

1) Sous-programme LINOM :

2) Objet : lecture du format des noms des individus, de leurs noms et du format d'édition d'une ligne du dessin de l'arbre.

3) Description des arguments :

3-1) . IENT INTEGER (E) : fichier de lecture.

3-2) . NIND INTEGER (E) : nombre d'individus.

3-3) . ICAR INTEGER (E) : nombre maximum de mots mémoire pour stocker le nom d'un individu.

3-4) . IFMTID INTEGER (S) : tableau de format variable.

3-5) . ITEM INTEGER (S) : tableau des noms des individus.

4) Sous-programme(s) requis :

aucun.

5) Sous-programme(s) appelant(s) :

appelé éventuellement (suivant les paramètres) par le sous-programme principal CLMIL.

6) Transmission des données :

* ni espaces communs, ni équivalences ;

* fichier utilisé en lecture : celui des paramètres IENT=41.

7) Divers :

néant.

1) Sous-programme INIT :

2) Objet : Initialisation à zéro du nombre de classes et initialisation du tableau LSYM facilitant l'adressage dans le tableau contenant la matrice triangulaire des similarités. Pour une ligne I, LSYM(I) contient le nombre de places occupées par les i-lèmes premiéres lignes dans le tableau unidimensionnel contenant la matrice triangulaire des similarités.

3) Description des arguments :

3-1) . N INTEGER (E) : nombre maximum de classes possibles.

3-2) . NCLTOT INTEGER (S) : nombre de classes.

3-3) . LSYM INTEGER (S) : tableau facilitant l'adressage.

4) Sous-programme(s) requis :

aucun.

5) Sous-programme(s) appelant(s) :

appelé par le sous-programme principal CLMIL.

6) Transmission des données :

ni espaces communs, ni équivalences, ni fichiers.

7) Divers :

néant.

1) Sous-programme CLAS1 :

2) Objet : ce sous-programme supervise la classification partielle de chaque tranche d'individus, pour chacune d'entre-elles, il appelle les sous-programmes de lecture/transposition des données et de classification partielle.

3) Description des arguments :

3-1) . IFMT INTEGER (E) : tableau contenant le format des données.

3-2) . IPARAM INTEGER (E) : tableau des paramètres.

3-3) . IFICH INTEGER (E) : tableau des numéros de fichiers.

3-4) . ICARDT INTEGER (E) : cardinal d'une tranche complète.

3-5) . NATT INTEGER (E) : nombre d'attributs.

3-6) . MXPAIR INTEGER (E) : nombre maximum de paires de classes agrégeables au même niveau de l'arbre.

3-7) . MXSYM INTEGER (E) : nombre maximum d'individus à classifier en même temps.

3-8) . NTOTAL INTEGER (E) : nombre maximum de classes permises.

3-9) . NIPOL INTEGER (E) : dimension du tableau de la représentation polonaise.

3-10) . MIQ INTEGER (E) : dimension du tableau de la matrice triangulaire inférieure des similarités.

3-11) . MIQ1 INTEGER (E) : dimension du tableau de travail pour la transposition des données.

3-12) . MIQ2 INTEGER (E) : dimension du tableau de la matrice triangulaire inférieure stricte des similarités.

3-13) . LSYM INTEGER (E) : tableau facilitant l'adressage des tables de similarités.

3-14) . NCLTOT INTEGER (ES) : nombre de classes.

3-15) . INDREP INTEGER (S) : tableau des représentants des classes.

3-16) . IORDRP INTEGER (S) : tableau assurant la correspondance entre les classes et leurs représentants après le tri de ces derniers.

3-17) . IX INTEGER (T) : tableau contenant le codage d'un individu lors de la transposition des données.

3-18) . IY INTEGER (T) : tableau contenant une colonne du fichier des données lors de sa transposition.

3-19) . IQ INTEGER (T) : tableau de travail pour la transposition des données.

3-20) . LECT INTEGER (T) : tableau de lecture d'une ligne du fichier transposé des données pour le calcul des indices de similarité.

3-21) . LADR INTEGER (T) : tableau permettant le repérage d'une classe lors de la construction de la représentation polonaise.

3-22) . IDBZON INTEGER (T) : tableau du début des classes lors de la construction de la représentation polonaise.

3-23) . IFNZON INTEGER (T) : tableau de la fin des classes lors de la construction de la représentation polonaise.

3-24) . ICDZON INTEGER (T) : tableau du cardinal des classes lors de la construction de la représentation polonaise.

3-25) . IPOL INTEGER (T) : tableau de la représentation polonaise.

3-26) . IPOL2 INTEGER (T) : tableau de travail lors de la construction de la représentation polonaise.

3-27) . IPAIR1 INTEGER (T) : pour chaque paire de classes agrégeables à un niveau, tableau indiquant la première classe.

3-28) . IPAIR2 INTEGER (T) : pour chaque paire de classes agrégeables à un niveau, tableau indiquant la deuxième classe.

3-29) . Q REAL (T) : tableau de la matrice triangulaire inférieure des similarités.

3-30) . Q2 REAL (T) : tableau de la matrice triangulaire inférieure stricte des similarités.

3-31) . XMU REAL (T) : tableau de la contribution des individus à la moyenne pour l'hypothèse d'absence de lien entre individus.

3-32) . SMU REAL (T) : tableau de la contribution des individus à la variance pour l'hypothèse d'absence de lien entre individus.

3-33) . CARRE REAL (T) : pour chaque individu, tableau de la somme des carrés des similarités avec chaque individu de sa classe.

3-34) . XNIV REAL (T) : tableau des indices des niveaux de l'arbre.

3-35) . NUMERR INTEGER (ER/ES) : code des erreurs.

4) Sous-programme(s) requis :

LCDON - CLS11 - OUVRE - FERME.

5) Sous-programme(s) appelant(s) :

appelé par le sous-programme principal CLMIL.

6) Transmission des données :

* ni espaces communs, ni équivalences;

* ouverture et fermeture (simulation) du fichier des données et du fichier temporaire de stockage des représentations polonaises partielles, mais pas d'utilisation directe de fichiers.

7) Divers :

néant.

1) Sous-programme LCDON :

2) Objet : recopie sur un fichier binaire de la partie des données concernant la tranche à classifier (partiellement) puis transposition de ce fichier binaire sur un autre fichier binaire.

3) Description des arguments :

3-1) . IFICH INTEGER (E) : tableau des numéros de fichiers.

3-2) . IFMT INTEGER (E) : tableau du format des données.

3-3) . M INTEGER (E) : dimension du tableau de travail.

3-4) . NATT INTEGER (E) : nombre d'attributs.

3-5) . MXSYM INTEGER (E) : nombre maximum de lignes à transposer.

3-6) . N INTEGER (E) : nombre de lignes à transposer.

3-7) . IQ INTEGER (T) : tableau de travail.

3-8) . IX INTEGER (T) : tableau de lecture d'une ligne du fichier non transposé.

3-9) . IY INTEGER (T) : tableau d'écriture sur le fichier transposé.

4) Sous-programme(s) requis :

ILIRE - IECRI - OUVRE - FERME.

5) Sous-programme(s) appelant(s) :

CLAS1.

6) Transmission des données :

- ni espaces communs, ni équivalences;
- fichiers utilisés :

  fichier des données : IFDON = IFICH(3)=11;

  fichier binaire des données de la tranche IFTAB = IFICH(4)=43;

fichier transposé : IFTABT = IFICH(5)=44.

7) Divers :

Algorithme de transposition :

Considérons le tableau de travail IQ de dimension M, il peut contenir N1=M/N lignes du fichier transposé à produire (ou colonnes du fichier actuel). Le fichier à produire va donc être rempli par groupe de N1 lignes (anciennes colonnes) contenues dans IQ. Pour remplir IQ, on lit l'ancien fichier ligne par ligne dans IX en recopiant dans IQ les colonnes de IX qui nous intéressent. Pour toute la transposition, on va lire N/N1 (ou N/N1+1) fois le fichier non transposé. Le tableau IY ne sert qu'à l'écriture sur le fichier transposé.

1) Sous-programme CLS11 :

2) Objet : classification partielle d'une tranche ; en fait ce sous-programme gère les sous-programmes des différentes phases de la classification (calcul des indices de similarité, réductions, construction de la représentation polonaise).

3) Description des arguments :

3-1) . IFICH INTEGER (E) : tableau des numéros des fichiers.

3-2) . IPARAM INTEGER (E) : tableau des paramètres.

3-3) . NATT INTEGER (E) : nombre d'attributs.

3-4) . MXSYM INTEGER (E) : nombre maximum d'individus à classifier en même temps.

3-5) . NTOTAL INTEGER (E) : nombre maximum de classes permises.

3-6) . NIPOL INTEGER (E) : dimension du tableau de la représentation polonaise.

3-7) . MIQ INTEGER (E) : dimension du tableau de la matrice triangulaire inférieure des similarités.

3-8) . MIQ2 INTEGER (E) : dimension du tableau de la matrice triangulaire inférieure stricte des similarités.

3-9) . MXPAIR INTEGER (E) : nombre maximum de paires de classes agrégeables à un même niveau de l'arbre.

3-10) . NT INTEGER (E) : numéro de la tranche à classifier.

3-11) . N INTEGER (E) : cardinal de la tranche à classifier.

3-12) . NCL INTEGER (E) : nombre de classes en dessous duquel on peut couper l'arbre.

3-13) . LSYM INTEGER (E) : tableau facilitant l'adressage des tables de similarités.

3-14) . NCLTOT INTEGER (ES) : nombre de classes.

3-15) . INDREP INTEGER (ES) : tableau des représentants des classes.

3-16) . IORDRP INTEGER (ES) : tableau assurant la correspondance entre les classes et leurs représentants après le tri de ces derniers.

3-17) . LECT INTEGER (T) : tableau de lecture d'une ligne du fichier transposé des données lors du calcul des indices de simiarités.

3-18) . LADR INTEGER (T) : tableau permettant de repérer les classes lors de la construction de la représentation polonaise.

3-19) . IDBZON INTEGER (T) : tableau indiquant le début des classes lors de la construction de la représentation polonaise.

3-20) . IFNZON INTEGER (T) : tableau indiquant la fin des classes lors de la construction de la représentation polonaise.

3-21) . ICDZON INTEGER (T) : tableau contenant le cardinal des classes lors de la construction de la représentation polonaise.

3-22) . IPOL INTEGER (T) : tableau de la représentation polonaise.

3-23) . IPOL2 INTEGER (T) : tableau de travail lors de la construction de la représentation polonaise.

3-24) . IPAIR1 INTEGER (T) : pour chaque paire de classes agrégeables au même niveau de l'arbre, tableau contenant la première classe.

3-25) . IPAIR2 INTEGER (T) : pour chaque paire de classes agrégeables au même niveau de l'arbre, tableau contenant la deuxième classe.

3-26) . Q REAL (T) : tableau de la matrice triangulaire inférieure des similarités.

3-27) . Q2 REAL (T) : tableau de la matrice triangulaire inférieure stricte des similarités.

3-28) . XMU REAL (T) : tableau de la contribution des individus à la moyenne pour l'hypothèse d'absence de

3-1

3-1

3-1

3-1

3-32) . NUMERR INTEGER (ER/ES) : code des erreurs.

4) Sous-programme(s) requis :

SIMIL - REDUC - SVSIM - NORMA - POLAR.

5) Sous-programme(s) appelant(s) :

CLAS1.

6) Transmission des données :

ni espaces communs, ni équivalences, ni fichiers utilisés.

7) Divers :

néant.

1) Sous-programme SIMIL :

2) Objet : calcul des indices de similarité centrés et réduits selon l'hypothèse d'absence de lien choisie.

3) Description des arguments :

3-1) . IFTABT INTEGER (E) : numéro du fichier transposé des données.

3-2) . ISORT INTEGER (E) : numéro du fichier de sortie.

3-3) . NATT INTEGER (E) : nombre d'attributs.

3-4) . MXSYM INTEGER (E) : nombre maximum d'individus à classifier en même temps.

3-5) . M INTEGER (E) : dimension du tableau de la matrice triangulaire inférieure des similarités.

3-6) . N INTEGER (E) : nombre d'individus à classifier.

3-7) . IHAL INTEGER (E) : hypothèse d'absence de lien choisie.

3-8) . LSYM INTEGER (E) : tableau facilitant l'adressage des tables de similarité.

3-9) . XMOY REAL (S) : moyenne des similarités.

3-10) . VARC REAL (S) : variance des similarités.

3-11) . XMAX REAL (S) : maximum des similarités.

3-12) . Q REAL (S) : tableau de la matrice triangulaire inférieure des similarités.

3-13) . LECT INTEGER (T) : tableau de lecture d'une ligne du fichier transposé des données.

3-14) . XMU REAL (T) : tableau de la contribution des individus à la moyenne pour l'hypothèse d'absence de lien entre individus.

3-15) . SMU REAL (T) : tableau de la contribution des individus à la variance pour l'hypothèse d'absence de lien entre individus.

3-16) . NUMERR INTEGER (ER/ES) : code des erreurs.

4) Sous-programme(s) requis :

OUVRE - FERME - ILIRE - ERROR.

5) Sous-programme(s) appelant(s) :

CLS11 - CLS22.

6) Transmission des données :

- ni espaces communs, ni équivalences;
- fichiers utilisés :
  - fichier de lecture de la table transposée des données : IFTABT=44;
  - fichier de sortie (erreur) pas directement utilisé : ISORT=42.

7) Divers :

* indice d'association entre les individus x et y :

$q(x,y) = (Nxy - u)/e$

avec : Nxy nombre d'attributs possédés conjointement par x et y;

Nz nombre d'attributs possédés par l'individu z;

N nombre total d'attributs.

$\overline{Nz} = N - Nz$

halp (IHAL=1) : $u = Nx*Ny/N$

$e = u^{1/2}$

halh (IHAL=2) : $u = Nx*Ny/n$

$e = (Nx*\overline{Nx}*Ny*\overline{Ny}/N^{2}*(N-1))^{1/2}$

* procédé de calcul :

L'indice peut en fait s'écrire sous la forme :

$q(x,y) = (Nxy-Ux*Uy)/Ex*Ey$ avec :

$Uz = Nz/N^{1/2}$ et

$Ez=Uz^{1/2}$ si on a choisi l'hypothèse d'absence de lien correspondant au modèle poissonien ou

$Ez=Nz*\overline{Nz}^{2}/(N(N-1))^{1/2}$ si on a choisi l'hypothèse d'absence de lien correspondant au modèle hypergéométrique.

* schéma de calcul :

(a) pour chaque individu z calcul de Nz dans le tableau XMU(*); calcul des indices bruts Nz1,z2 pour toute paire d'individus (z1,z2);

(b) calcul des Uz dans le tableau XMU(*) et des Ez dans le tableau SMU(*);

(c) centrage et réduction suivant l'hypothèse d'absence de lien choisie.

remarque : le maximum des indices, leur moyenne et leur variance sont calculés de proche en proche lors de l'étape (c).

1) Sous-programme REDUC :

2) Objet : réduction globale des indices de similarité (au choix homothétie de rapport lambda ou centrage et réduction globaux); calcul des termes B1, B2 et B3 nécessaires au calcul de l'indice des niveaux.

$$B1 = \left(\sum_{\substack{1<=i<=n \\ i \neq j}} Pij \right)^2$$

$$B2 = \sum_{1<=i<=n} \left(\sum_{\substack{1<=j<=n \\ i \neq j}} Pij \right)^2$$

$$B3 = \sum_{\substack{1<=j<=n \\ i \neq j}} Pij^2$$

avec n : nombre d'individus;
Pij : indice de similarité entre les individus i et j (indice réduit globalement).

3) Description des arguments :

3-1) . MXSYM INTEGER (E) : nombre maximum d'individus à classifier en même temps.

3-2) . M INTEGER (E) : dimension du tableau de la matrice triangulaire inférieure des similarités.

3-3) . M2 INTEGER (E) : dimension du tableau de la matrice triangulaire inférieure stricte des similarités.

3-4) . N INTEGER (E) : nombre d'individus à classifier.

3-5) . IREDUC INTEGER (E) : réduction globale choisie.

3-6) . XMOY REAL (E) : moyenne des similarités.

3-7) . VARC REAL (E) : racine carrée de la variance des similarités.

3-8) . XMAX REAL (E) : maximum des similarités.

3-9) . LSYM INTEGER (E) : tableau facilitant l'adressage des tables de similarités.

3-10) . Q REAL (ES) : tableau de la matrice triangulaire inférieure des similarités.

3-11) . B1 REAL (S) : terme B1 décrit plus haut.

3-12) . B2 REAL (S) : terme B2 décrit plus haut.

3-13) . B3 REAL (S) : terme B3 décrit plus haut.

3-14) . Q2 REAL (S) : tableau de la matrice triangulaire inférieure stricte des similarités.

4) Sous-programme(s) requis :

aucun.

5) Sous-programme(s) appelant(s) :

CLS11 - CLS22.

6) Transmission des données :

ni espaces communs, ni équivalences, ni fichiers.

7) Divers :

On sauvegarde la matrice des similarités diagonale exclue dans le tableau Q2 afin de pouvoir calculer l'indice des niveaux de l'arbre, le tableau Q étant modifié pendant la construction de la représentation polonaise.

1) Sous-programme SVSIM :

2) Objet : édition sur fichier pour une classification (partielle ou globale) de la moyenne et l'écart type ou seulement du maximum des similarités d'une part et de la table des similarités d'autre part.

3) Description des arguments :

3-1) . IFSIM INTEGER (E) : numéro du fichier de sortie.

3-2) . IREDUC INTEGER (E) : réduction globale choisie.

3-3) . N INTEGER (E) : nombre d'individus classifiés.

3-4) . MIQ INTEGER (E) : dimension du tableau de la matrice triangulaire inférieure des similarités.

3-5) . XMOY REAL (E) : moyenne des similarités.

3-6) . VARC REAL (E) : écart type des similarités.

3-7) . XMAX REAL (E) : maximum des similarités.

3-8) . Q REAL (E) : tableau de la matrice triangulaire inférieure des similarités.

4) Sous-programme(s) requis :

aucun.

5) Sous-programme(s) appelant(s) :

CLS11 - CLS22.

6) Transmission des données :

- ni espaces communs, ni équivalences;
- fichier utilisé en sortie : IFSIM=61.

7) Divers :

néant.

1) Sous-programme NORMA :

2) Objet : application de la fonction de répartition de la loi normale à la table des similarités puis passage à la forme -log(-log(.)).

3) Description des arguments :

3-1) . N INTEGER (E) : nombre d'individus classifiés.

3-2) . MIQ INTEGER (E) : dimension du tableau de la matrice triangulaire inférieure des similarités.

3-3) . Q REAL (ES) : tableau de la matrice triangulaire inférieure des similarités.

4) Sous-programme(s) requis :

aucun.

5) Sous-programme(s) appelant(s) :

CLS11 - CLS22.

6) Transmission des données :

ni espaces communs, ni équivalences, ni fichiers.

7) Divers :

néant.

1) Sous-programme POLAR :

2) Objet : construction de la représentation polonaise de l'arbre de la classification partielle d'une tranche d'individus.

3) Description des arguments :

3-1) . IFICH INTEGER (E) : tableau des numéros des fichiers.

3-2) . MXSYM INTEGER (E) : nombre maximum d'individus à classifier en même temps.

3-3) . NTOTAL INTEGER (E) : nombre maximum de classes permises.

3-4) . MXPAIR INTEGER (E) : nombre maximum de paires de classes agrégeables à un niveau de l'arbre.

3-5) . NIPOL INTEGER (E) : dimension du tableau de la représentation polonaise.

3-6) . MIQ INTEGER (E) : dimension du tableau de la matrice triangulaire inférieure des similarités.

3-7) . MIQ2 INTEGER (E) : dimension du tableau de la matrice triangulaire inférieure stricte des similarités.

3-8) . N INTEGER (E) : nombre d'individus à classifier.

3-9) . NCL INTEGER (E) : nombre de classes en-dessous duquel on peut couper l'arbre (arrêter la classification).

3-10) . NT INTEGER (E) : numéro de la classe.

3-11) . ICARDT INTEGER (E) : cardinal d'une tranche complète.

3-12) . B1 REAL (E) : voir REDUC.

3-13) . B2 REAL (E) : voir REDUC.

3-14) . B3 REAL (E) : voir REDUC.

3-15) . LSYM INTEGER (E) : tableau facilitant l'adressage des tables de similarités.

3-16) . Q2 REAL (E) : tableau de la matrice triangulaire inférieure stricte des similarités.

3-17) . NCLTOT INTEGER (ES) : nombre de classes.

3-18) . INDREP INTEGER (ES) : tableau des représentants des classes.

3-19) . IORDRP INTEGER (ES) : tableau assurant la correspondance entre les classes et leurs représentants après le tri de ces derniers.

3-20) . Q REAL (ES) : tableau de la matrice triangulaire inférieure des similarités.

3-21) . LADR INTEGER (T) : tableau permettant l'adressage d'une classe lors de la construction de la représentation polonaise.

3-22) . IPOL INTEGER (T) : tableau de la représentation polonaise.

3-23) . IPOL2 INTEGER (T) : tableau de travail.

3-24) . IDBZON INTEGER (T) : tableau du début des classes lors de la construction de la représentation polonaise.

3-25) . IFNZON INTEGER (T) : tableau de la fin des classes lors de la construction de la représentation polonaise.

3-26) . ICDZON INTEGER (T) : tableau contenant le cardinal des classes lors de la construction de la représentation polonaise.

3-27) . IPAIR1 INTEGER (T) : pour chaque paire de classes agrégeables à un niveau de l'arbre, tableau indiquant la première classe.

3-28) . IPAIR2 INTEGER (T) : pour chaque paire de classes agrégeables à un niveau de l'arbre, tableau indiquant la deuxième classe.

3-29) . XNIV REAL (T) : tableau des indices des niveaux de l'arbre.

3-30) . CARRE REAL (T) : pour chaque individu, tableau de la somme des carrés des similarités avec chaque individu de sa classe.

3-31) . NUMERR INTEGER (ER/ES) : code des erreurs.

4) Sous-programme(s) requis :

PRAGR - POLON - MAJR - ACTUA - STNIV - STOCK.

5) Sous-programme(s) appelant(s) :

CLS11.

6) Transmission des données :

ni espaces communs, ni équivalences, ni fichiers.

7) Divers :

* niveau de coupure - arrêt de la classification partielle :

L'argument NCL indique le nombre de classes en-dessous duquel on peut couper l'arbre; en fait on continue la classification partielle tant que le nombre de classes est supérieur à NCL-5. Le niveau de coupure sera le niveau dont l'indice est maximum et pour lequel le nombre de classes est compris entre NCL-5 et NCL.

* algorithme :

(a) initialisations - en particulier tout individu forme une classe, niveau=0.

(b) recherche des NPAIR paires de classes agrégeables (sous-programme PRAGR), première classe dans IPAIR1, deuxième dans IPAIR2.

(c) niveau=niveau+1,
pour chaque paire de classes agrégeables

(c1) agrégation - mise à jour de la représentation polonaise (sous-programme POLON).

(c2) réactualisation des indices de similarité (sous-programme MAJR).

(c3) réactualisation de la numérotation des classes (sous-programme ACTUA).

(c4) si nombre de classes compris entre NCL-5 et NCL, calcul de l'indice de niveau et éventuellement réactualisation du niveau de coupure.

(d) si nombre de classes supérieur à NCL-5 alors aller à (b).

(e) détermination d'un représentant par classe, stockage sur un fichier temporaire de la représentation polonaise des classes (sous-programme STOCK).

(f) fin du sous-programme.

* structure des données :

a) représentation polonaise :

Les représentations polonaises des classes sont rangées dans le tableau IPOL. Pour chaque classe, le tableau IDBZON donne l'indice du début de la représentation polonaise de cette classe dans IPOL, et le tableau IFNZON l'indice de fin. De plus le tableau ICDZON contient le cardinal de la classe.

b) modification de la stucture de données après l'agrégation de deux classes K1 et K2 (K2>K1) :

On met bout à bout dans IPOL les représentations polonaises des classes K1 et K2 précédées de (-niveau d'agrégation). La nouvelle classe sera désignée K1, l'ancienne classe K2 disparaît et on réactualise les tableaux IDBZON,IFNZON et ICDZON. Les autres classes numérotées Ki avec Ki>K2 sont renumérotées Ki-1. Ceci implique de pouvoir retrouver l'adresse dans le tableau Q des indices de similarité, ce qui est assuré par le tableau LADR.
Au début, l'élément de ce tableau correspondant à une classe donnée est le numéro de cette classe. Lorsque au cours d'une agrégation une classe Kj disparaît, toutes les classes suivantes voient leur numéro diminuer de un, pour ces classes on réactualise le tableau LADR (l'adresse d'une classe ne change pas quand on change son numéro). En réalité, le tableau LADR ne donne que le "numéro primitif" des classes.

1) Sous-programme PRAGR :

2) Objet : déterminer la (les) paire(s) de classes dont l'indice de similarité est maximum.

3) Description des arguments :

3-1) . ISORT INTEGER (E) : numéro du fichier de sortie (erreur).

3-2) . MXSYM INTEGER (E) : nombre maximum d'individus à classifier en même temps.

3-3) . MIQ INTEGER (E) : dimension du tableau de la matrice triangulaire inférieure des similarités.

3-4) . MXPAIR INTEGER (E) : nombre maximum de paires agrégeables.

3-5) . NBCLAS INTEGER (E) : nombre courant de classes.

3-6) . LSYM INTEGER (E) : tableau facilitant l'adressage des tables de similarités.

3-7) . LADR INTEGER (E) : tableau permettant le repérage d'une classe.

3-8) . Q REAL (E) : tableau de la matrice triangulaire inférieure des similarités.

3-9) . NPAIR INTEGER (S) : nombre de paire(s) de classes agrégeables.

3-10) . IPAIR1 INTEGER (S) : pour chaque paire de classes agrégeables, tableau indiquant la première classe.

3-11) . IPAIR2 INTEGER (S) : pour chaque paire de classes agrégeables, tableau indiquant la deuxième classe.

3-12) . NUMERR INTEGER (ER/ES) : code des erreurs.

4) Sous-programme(s) requis :

ERROR.

5) Sous-programme(s) appelant(s) :

POLAR - POLGL.

6) Transmission des données :

ni espaces communs, ni équivalences, ni fichiers directement utilisés.

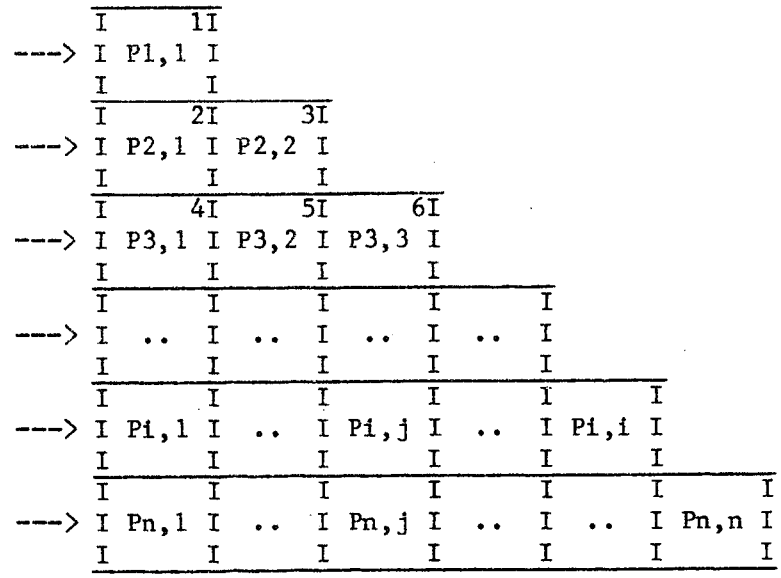
7) Divers :

* rangement des paires de classes agrégeables :

Soit (K1,K2) (avec K2>K1) une paire de classes agrégeables (c'est à dire d'indice de similarité maximum) : K1 sera rangée dans le tableau IPAIR1 et K2 dans le tableau IPAIR2; la variable NPAIR indiquant le nombre de paires de classes agrégeables.

* Accès au tableau de la matrice triangulaire des similarités :

On a vu (POLAR/Divers) que le numérotage des classes change après une agrégation, c'est le tableau LADR qui contient les numéros primitifs des classes.
De plus, la matrice triangulaire des similarités est rangée dans un tableau ligne de la manière suivante :



l'indice d'un élément Pi,j est alors (i>=j) : i(i-1)/2+j.

Pour une ligne l donnée, le tableau LSYM a été initialisé (sous-programme INIT) au nombre d'éléments du tableau ligne Q précédant le premier élément de cette ligne, c'est à dire l(l-1)/2.

l'adresse d'un élément Pi,j devient LSYM(i)+j avec :
* i>=j et
* i et j étant les "numéros primitifs" des classes (donnés par le tableau LADR.

L'adresse de PI,J où I et J sont les numéros actuels des classes est :
LSYM(LADR(I))+LADR(J) ; le programme conservant l'ordre des classes :
I>J ==> LADR(I)>LADR(J).

1) Sous-programme POLON :

2) Objet : mise à jour de la représentation polonaise de l'arbre lors de l'agrégation de deux classes; pour ce faire on met bout à bout les représentations polonaises des deux classes, le tout précédé de moins le niveau d'agrégation dans le tableau IPOL.

3) Description des arguments :

3-1) . MXSYM INTEGER (E) : nombre maximum d'individus à classifier en même temps.

3-2) . NIPOL INTEGER (E) : dimension du tableau de la représentation polonaise.

3-3) . NBCLAS INTEGER (E) : nombre courant de classes.

3-4) . K1 INTEGER (E) : première des deux classes à agréger.

3-5) . K2 INTEGER (E) : deuxième des deux classes à agréger.

3-6) . NIV INTEGER (E) : niveau de l'agrégation.

3-7) . IPOL INTEGER (ES) : tableau de la représentation polonaise.

3-8) . IDBZON INTEGER (ES) : tableau repérant le début des classes dans le tableau IPOL.

3-9) . IFNZON INTEGER (ES) : tableau repérant la fin des classes dans le tableau IPOL.

3-10) . IPOL2 INTEGER (T) : tableau de travail utilisé pour faire des sauvegardes.

4) Sous-programme(s) requis :

IDECD.

5) Sous-programme(s) appelant(s) :

POLAR - POLGL.

6) Transmission des données :

ni espaces communs, ni équivalences, ni fichiers.

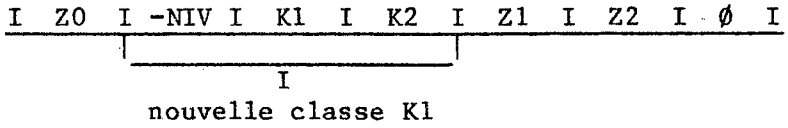
7) Divers :

* schéma initial du tableau IPOL :

I Z0 I K1 I Z1 I K2 I Z2 I Ø I

où K1 et K2 sont les classes à agréger au niveau NIV.

* schéma du tableau IPOL après agrégation :



* algorithme général :

(a) décalage de Z2 d'une case vers la droite.

(b) sauvegarde de K2 dans le tableau IPOL2.

(c) décalage de la zone Z1 de k2+1 cases vers la droite où k2 est la longueur de K2.

(d) recopie de la zone K2 avant la zone Z1 dans le tableau IPOL.

(e) décalage de la zone K1 d'une case vers la droite.

(f) écriture de -NIV avant la nouvelle zone K1 dans IPOL.

* remarque : simplification lorsque Z1 et/ou Z2 est vide.

1) Sous-programme IDECD :

2) Objet : décalage d'une partie des éléments d'un tableau vers la droite.

3) Description des arguments :

3-1) . ID INTEGER (E) : indice du premier élément de la zone à décaler.

3-2) . IFIN INTEGER (E) : indice du dernier élément de la zone à décaler.

3-3) . IL INTEGER (E) : amplitude du décalage.

3-4) . IDIMS INTEGER (E) : dimension du tableau.

3-5) . ITAB INTEGER (ES) : nom du tableau.

4) Sous-programme(s) requis :

aucun.

5) Sous-programme(s) appelant(s) :

POLON.

6) Transmission des données :

ni espaces communs, ni équivalences, ni fichiers.

7) Divers :

néant.

1) Sous-programme MAJR :

2) Objet : réactualisation de la matrice des similarités après une agrégation entre les classes I et J.
Elle a lieu uniquement sur les indices de similarité entre la nouvelle classe et les anciennes classes non concernées par l'agrégation - formule :
$PIJ,K=Max[(PI,K)+\log(|I|)),(PJ,K+\log(|J|))] - \log(|I|+|J|)$

3) Description des arguments :

3-1) . MXSYM INTEGER (E) : nombre maximum d'individus à classifier en même temps.

3-2) . MIQ INTEGER (E) : dimension du tableau de la matrice triangulaire inférieure des similarités.

3-3) . K1 INTEGER (E) : première des deux classes agrégées.

3-4) . K2 INTEGER (E) : deuxième des deux classes agrégées.

3-5) . LSYM INTEGER (E) : tableau facilitant l'adressage des tables de similarités.

3-6) . NBCLAS INTEGER (ES) : nombre de classes.

3-7) . LADR INTEGER (ES) : tableau permettant le repérage des classes.

3-8) . ICDZON INTEGER (ES) : tableau du cardinal des classes.

3-9) . Q REAL (ES) : tableau de la matrice triangulaire inférieure des similarités.

4) Sous-programme(s) requis :

aucun.

5) Sous-programme(s) appelant(s) :

POLAR - POLGL.

6) Transmission des données :

ni espaces communs, ni équivalences, ni fichiers.

7) Divers :

Même technique d'adressage que pour PRAGR.

1) Sous-programme ACTUA :

2) Objet : renumérotation de classes après une agrégation.

3) Description des arguments :

3-1) . MXPAIR INTEGER (E) : nombre maximum de paires de classes agrégeables à un niveau de l'arbre.

3-2) . NPAIR INTEGER (E) : nombre de paires de classes agrégeables au niveau courant.

3-3) . IPAIR INTEGER (E) : indice de la paire de classes que l'on vient d'agréger.

3-4) . K1 INTEGER (E) : première des deux classes que l'on vient d'agréger.

3-5) . K2 INTEGER (E) : deuxième des deux classes que l'on vient d'agréger.

3-6) . IPAIR1 INTEGER (ES) : pour chaque paire de classes agrégeables, tableau indiquant la première classe.

3-7) . IPAIR2 INTEGER (ES) : pour chaque paire de classes agrégeables, tableau indiquant la deuxième classe.

4) Sous-programme(s) requis :

aucun.

5) Sous-programme(s) appelant(s) :

POLAR - POLGL.

6) Transmission des données :

ni espaces communs, ni équivalences, ni fichiers.

7) Divers :

néant.

1) Sous-programme STNIV :

2) Objet : calcul de l'indice des niveaux (statistique des niveaux) de l'arbre à un niveau donné.

indice brut : $SBniv = \sum_{\substack{1<=i,j<=n \\ i\neq j}} Rij*Pij$

où n est le nombre d'individus classifiés;
Pij est l'indice de similarité entre les individus i et j;
Rij = 1 si au niveau courant i et j appartiennent à la même classe,
Rij = 0 sinon.

moyenne : $u = (A1*B1)^{1/2} /n*(n-1)$

variance : $e^2 = -A1*B1/(n*(n-1)) + 2*A3*B3/n*(n-1) + 4*(A2-A3)*(B2-B3)/n*(n-1)*(n-2) + (A1-4*A2+2*A3)*(B1-4*B2+2*B3)/(n*(n-1)*(n-2)*(n-3))^2.$

où :

$$A1 = \left(\sum_{\substack{1<=i<=n \\ i\neq j}} Rij \right)^2$$

$$A2 = \sum_{1<=i<=n} \left(\sum_{\substack{1<=j<=n \\ i\neq j}} Rij \right)^2$$

$$A3 = \sum_{\substack{1<=j<=n \\ i\neq j}} Rij^2$$

$$B1 = \left(\sum_{\substack{1<=i<=n \\ i \neq j}} Pij \right)^2$$

$$B2 = \sum_{1<=i<=n} \left(\sum_{\substack{1<=j<=n \\ i \neq j}} Pij \right)^2$$

$$B3 = \sum_{\substack{1<=j<=n \\ i \neq j}} Pij^2$$

indice définitif : $$Sniv = \frac{SBniv - u}{e}$$

3) Description des arguments :

3-1) . NELEMT INTEGER (E) : nombre d'individus classifiés avec les tranches précédantes (pour l'adressage).

3-2) . MXSYM INTEGER (E) : nombre maximum d'individus à classifier en même temps.

3-3) . MIQ2 INTEGER (E) : dimension du tableau de la matrice triangulaire inférieure stricte des similarités.

3-4) . NIPOL INTEGER (E) : dimension du tableau de la représentation polonaise.

3-5) . N INTEGER (E) : nombre d'individus à classifier.

3-6) . NIV INTEGER (E) : niveau courant de l'arbre.

3-7) . NBCLAS INTEGER (E) : nombre de classes.

3-8) . LSYM INTEGER (E) : tableau facilitant l'adressage des tables de similarités.

3-9) . Q2 REAL (E) : tableau de la matrice triangulaire inférieure stricte des similarités.

3-10) . IPOL INTEGER (E) : tableau de la représentation polonaise.

3-11) . IDBZON INTEGER (E) : tableau repérant le début des classes dans le tableau IPOL.

3-12) . IFNZON INTEGER (E) : tableau repérant la fin des classes dans le tableau IPOL.

3-13) . ICDZON INTEGER (E) : tableau contenant les cardinaux des classes.

3-14) . B1 REAL (E) : voir plus haut.

3-15) . B2 REAL (E) : voir plus haut.

3-16) . B3 REAL (E) : voir plus haut.

3-17) . XNIV REAL (ES) : tableau des indices des niveaux.

4) Sous-programme(s) requis :

aucun.

5) Sous-programme(s) appelant(s) :

POLAR - POLGL.

6) Transmission des données :

ni espaces communs, ni équivalences, ni fichiers.

7) Divers :

* calcul de l'indice brut :

il s'agit en fait de la somme pour chaque classe de la somme des indices de similarité entre individus de la classe.

* déjà calculés :

on s'aperçoit que les termes B1,B2 et B3 sont constants lors de la classification d'une même tranche d'individus : ils ne sont donc calculés qu'une fois par tranche par le sous-programme REDUC.

* termes A1,A2 et A3 :

* Rij étant égal soit à 0, soit à 1 on a : $A1 = A3^2$

* deux individus sont réunis si et seulement si ils appartiennent à la même classe, pour avoir A2 et A3, il suffit donc de faire la somme de la contribution de chaque classe à A2 et à A3.

* contribution d'une classe (de cardinal c) à A2 :

$$\sum_{1<=i<=c} \Big(\sum_{\substack{1<=j<=c \\ j\neq i}} 1 \Big)^2 = \sum_{1<=i<=c} (c-1) = c((c-1)^2)$$

* contribution d'une classe (de cardinal c) à A3 :

$$\sum_{\substack{1<=i,j<=c \\ i\neq j}} 1 = c(c-1).$$

* tableau Q2 :

le tableau Q étant modifié lors du calcul de le représentation polonaise, il a été nécessaire pour calculer cet indice d'en faire une sauvegarde dans le tableau Q2 (sous-programme REDUC). La diagonale n'a pas été sauvegardée. L'adressage de Q2 est basé sur la même technique que l'adressage de Q.

1) Sous-programme STOCK :

2) Objet : détermination d'un représentant par classe : l'individu faisant le maximum de la somme des carrés des similarités dans sa classe; et pour chaque classe écriture sur un fichier temporaire de sa représentation polonaise précédée de sa longueur.

Note : on connecte au tableau des représentants un tableau contenant un codage du numéro de la classe et de la tranche pour assurer la correspondance entre les classes et leurs représentants après le tri de ces derniers.

3) Description des arguments :

3-1) . ITEMP1 INTEGER (E) : numéro du fichier temporaire.

3-2) . MXSYM INTEGER (E) : nombre maximum d'individus à classifier en même temps.

3-3) . NIPOL INTEGER (E) : dimension du tableau de la représentation polonaise.

3-4) . NTOTAL INTEGER (E) : nombre maximum de classes permises.

3-5) . MIQ2 INTEGER (E) : dimension du tableau de la matrice triangulaire inférieure stricte des similarités.

3-6) . N INTEGER (E) : nombre d'individus classifiés.

3-7) . ICARDT INTEGER (E) : cardinal d'une tranche complète.

3-8) . NT INTEGER (E) : numéro de la tranche.

3-9) . NVCOUP INTEGER (E) : niveau de coupure.

3-10) . LSYM INTEGER (E) : tableau facilitant l'adressage des tables de similarités.

3-11) . Q2 REAL (E) : tableau de la matrice triangulaire inférieure stricte des similarités.

3-12) . NCLTOT INTEGER (ES) : nombre de classes.

3-13) . INDREP INTEGER (ES) : tableau des représentants.

3-14) . IORDRP INTEGER (ES) : tableau assurant la correspondance entre les classes et leurs représentants après le tri de ces derniers (connecté au tableau des

représentants).

3-15) . IPOL INTEGER (ES) : tableau de la représentation polonaise.

3-16) . IPOL2 INTEGER (T) : tableau de travail.

3-17) . CARRE REAL (T) : tableau de la somme des carrés des similarités entre individus d'une même classe.

4) Sous-programme(s) requis :

SOMMC - IECRI.

5) Sous-programme(s) appelant(s) :

POLAR.

6) Transmission des données :

- ni espaces communs, ni équivalences.
- fichier temporaire utilisé en écriture : ITEMP1=45.

7) Divers :

* les représentations polonaises précédées de leur longueur sont toutes au fur et à mesure stockées dans le tableau IPOL2 qui est déversé en une seule fois dans le fichier temporaire.
* à la lecture de chaque classe, le début du tableau IPOL est écrasé, on y range les individus de la classe.
* pour N individus partitionnés en classes, la somme des longueurs des représentations polonaises des classes précédées de leur longueur est égale à 2*N.

1) Sous-programme SOMMC :

2) Objet : détermine le représentant d'une classe : individu de la classe réalisant le maximum pour la somme des carrés des similarités avec les autres individus de la classe.

3) Description des arguments :

3-1) . NINDCL INTEGER (E) : nombre d'individus de la classe.

3-2) . MXSYM INTEGER (E) : nombre maximum d'individus à classifier en même temps.

3-3) . NIPOL INTEGER (E) : dimension du tableau de la représentation polonaise.

3-4) . MIQ2 INTEGER (E) : dimension du tableau de la matrice triangulaire inférieure stricte des similarités.

3-5) . IPOL INTEGER (E) : tableau de la représentation polonaise, en fait au début du tableau suite des individus de la classe.

3-6) . LSYM INTEGER (E) : tableau facilitant l'adressage des tables de similarités.

3-7) . Q2 REAL (E) : tableau de la matrice triangulaire inférieure stricte des similarités.

3-8) . IMXCAR INTEGER (S) : représentant de la classe.

3-9) . CARRE REAL (T) : pour chaque individu tableau de la somme des carrés des similarités avec chaque individu de sa classe.

4) Sous-programme(s) requis :

aucun.

5) Sous-programme(s) appelant(s) :

STOCK.

6) Transmission des données :

ni espaces communs, ni équivalences, ni fichiers.

7) Divers :

néant.

1) Sous-programme TRIRP :

2) Objet : tri par ventilation du tableau des représentants (en permutant aussi à chaque fois le tableau assurant la correspondance représentants-classes) afin d'optimiser la relecture du fichier des données pour la classification globale.

3) Description des arguments :

3-1) . IPARAM INTEGER (E) : tableau des paramètres.

3-2) . NTOTAL INTEGER (E) : nombre maximum de classes permises.

3-3) . NCLTOT INTEGER (E) : nombre total de classes.

3-4) . ICARDT INTEGER (E) : cardinal d'une tranche complète.

3-5) . INDREP INTEGER (ES) : tableau des représentants.

3-6) . IORDRP INTEGER (ES) : tableau assurant la correspondance entre les classes et leurs représentants.

3-7) . ICHAMP INTEGER (T) : tableau de travail connecté au tableau logique de présence-absence des représentants.

3-8) . PRESEN LOGICAL (T) : tableau de présence-absence des représentants.

4) Sous-programme(s) requis :

aucun.

5) Sous-programme(s) appelant(s) :

appelé par le sous-programme principal CLMIL.

6) Transmission des données :

ni espaces communs, ni équivalences, ni fichiers.

7) Divers :

* de par le découpage en tranches, le numéro d'un représentant d'une tranche T1 est inférieur au numéro d'un représentant d'une tranche T2 si T2>T1, par conséquent, pour faire le tri total, il suffit de trier les représentants tranche par tranche.

1) Sous-programme CLAS2 :

2) Objet : gère la classification globale (des représentants), appelle les sous-programmes de lecture et transposition du codage des représentants et de classification globale des représentants.

3) Description des arguments :

3-1) . IFMT INTEGER (E) : tableau du format de lecture des données.

3-2) . IPARAM INTEGER (E) : tableau des paramètres.

3-3) . IFICH INTEGER (E) : tableau des numéros des fichiers.

3-4) . NATT INTEGER (E) : nombre d'attributs.

3-5) . MXSYM INTEGER (E) : nombre maximum d'individus à classifier en même temps.

3-6) . NCLTOT INTEGER (E) : nombre de représentants à classifier.

3-7) . NTOTAL INTEGER (E) : nombre maximum de classes permises.

3-8) . MXPAIR INTEGER (E) : nombre maximum de paires de classes agrégeables à un niveau donné.

3-9) . NIPOL INTEGER (E) : dimension du tableau de la représentation polonaise.

3-10) . MIQ INTEGER (E) : dimension du tableau de la matrice triangulaire inférieure des similarités.

3-11) . MIQ1 INTEGER (E) : dimension du tableau de travail pour la transposition du fichier des données.

3-12) . MIQ2 INTEGER (E) : dimension du tableau de la matrice triangulaire inférieure stricte des similarités.

3-13) . LSYM INTEGER (E) : tableau facilitant l'adressage des tables de similarités.

3-14) . INDREP INTEGER (E) : tableau des représentants.

3-15) . NBNVS INTEGER (S) : nombre de niveaux significatifs.

3-16) . IPOL INTEGER (S) : tableau de la représentation polonaise.

3-17) . NVSG INTEGER (S) : tableau des niveaux significatifs.

3-18) . IX INTEGER (T) : tableau contenant le codage d'un représentant lors de la transposition des données.

3-19) . IY INTEGER (T) : tableau d'écriture sur le fichier transposé des données.

3-20) . IQ INTEGER (T) : tableau de travail pour la transposition des données.

3-21) . LECT INTEGER (T) : tableau de lecture d'une ligne du fichier transposé des données lors du calcul des indices de similarité.

3-22) . LADR INTEGER (T) : tableau permettant le repérage des classes lors de la construction de la représentation polonaise.

3-23) . IDBZON INTEGER (T) : tableau contenant le début des classes lors de la construction de la représentation polonaise.

3-24) . IFNZON INTEGER (T) : tableau contenant la fin des classes lors de la construction de la représentation polonaise.

3-25) . ICDZON INTEGER (T) : tableau contenant le cardinal des classes lors de la construction de la représentation polonaise.

3-26) . IPOL2 INTEGER (T) : tableau de travail lors de la construction de la représentation polonaise.

3-27) . IPAIR1 INTEGER (T) : pour chaque paire de classes agrégeables à un niveau donné, tableau contenant la première classe.

3-28) . IPAIR2 INTEGER (T) : pour chaque paire de classes agrégeables à un niveau donné, tableau contenant la deuxième classe.

3-29) . Q REAL (T) : tableau de la matrice triangulaire inférieure des similarités.

3-30) . Q2 REAL (T) : tableau de la matrice triangulaire inférieure stricte des similarités.

3-31) . XMU REAL (T) : tableau de la contribution des représentants à la moyenne pour l'hypothèse d'absence de lien entre représentants.

3-32) . SMU REAL (T) : tableau de la contribution des représentants à la variance pour l'hypothèse d'absence de lien entre représentants.

3-33) . XNIV REAL (T) : tableau des indices des niveaux.

3-34) . NUMERR INTEGER (ER/ES) : code des erreurs.

4) Sous-programme(s) requis :

LCDNG - CLS22.

5) Sous-programme(s) appelant(s) :

appelé par le sous-programme principal CLMIL.

6) Transmission des données :

ni espaces communs, ni équivalences, ni fichiers.

7) Divers :

Ce sous-programme supervise la classification globale (des représentants des classes), il est symétrique au sous-programme CLAS1 qui lui supervise la classification partielle des tranches d'individus.

1) Sous-programme LCDNG :

2) Objet : relit entièrement le fichier des données et écrit sur un fichier binaire le codage des représentants des classes puis transpose ce fichier sur un autre fichier binaire.

3) Description des arguments :

3-1) . IFICH INTEGER (E) : tableau des numéros de fichiers.

3-2) . IFMT INTEGER (E) : tableau du format des données.

3-3) . MIQ1 INTEGER (E) : dimension du tableau de travail IQ.

3-4) . NATT INTEGER (E) : nombre d'attributs.

3-5) . MXSYM INTEGER (E) : nombre maximum d'individus à classifier en même temps.

3-6) . NTOTAL INTEGER (E) : nombre maximum de classes permises.

3-7) . NCLTOT INTEGER (E) : nombre de représentants.

3-8) . INDREP INTEGER (E) : tableau des représentants.

3-9) . IQ INTEGER (T) : tableau de travail.

3-10) . IX INTEGER (T) : tableau de lecture d'une ligne du fichier non transposé.

3-11) . IY INTEGER (T) : tableau d'écriture dans le fichier transposé.

4) Sous-programme(s) requis :

ILIRE - IECRI - OUVRE - FERME.

5) Sous-programme(s) appelant(s) :

CLAS2.

6) Transmission des données :

- ni espaces communs, ni équivalences.

- fichiers utilisés :

  fichier des données : IFDON = IFICH(3) = 11,

  fichier binaire du codage des représentants : IFTAB=IFICH(4)=43,

  fichier transposé : IFTABT = IFICH(5) = 44.

7) Divers :

* pour la technique de transposition voir LCDON.

1) Sous-programme CLS22 :

2) Objet : classification totale des représentants des classes (gestion des sous-programmes des différentes phases), sous-programme symétrique à CLS11.

3) Description des arguments :

3-1) . IFICH INTEGER (E) : tableau des numéros de fichiers.

3-2) . IPARAM INTEGER (E) : tableau des paramètres.

3-3) . MXSYM INTEGER (E) : nombre maximum d'individus à classifier en même temps.

3-4) . NATT INTEGER (E) : nombre d'attributs.

3-5) . NIPOL INTEGER (E) : dimension du tableau de la représentation polonaise.

3-6) . NCLTOT INTEGER (E) : nombre de représentants à classifier.

3-7) . MIQ INTEGER (E) : dimension du tableau de la matrice triangulaire inférieure des similarités.

3-8) . MIQ2 INTEGER (E) : dimension du tableau de la matrice triangulaire inférieure stricte des similarités.

3-9) . MXPAIR INTEGER (E) : nombre maximum de paires de classes agrégeables à un niveau de l'arbre.

3-10) . LSYM INTEGER (E) : tableau facilitant l'adressage des tables de similarités.

3-11) . NBNVS INTEGER (S) : nombre de niveaux significatifs.

3-12) . IPOL INTEGER (S) : tableau de la représentation polonaise.

3-13) . NVSG INTEGER (S) : tableau des niveaux significatifs.

3-14) . LECT INTEGER (T) : tableau de lecture des données pour le calcul des indices de similarité.

3-15) . LADR INTEGER (T) : tableau permettant le repérage des classes lors de la construction de la représentation polonaise.

3-16) . IDBZON INTEGER (T) : tableau contenant le début des classes lors de la construction de la représentation polonaise.

3-17) . IFNZON INTEGER (T) : tableau contenant la fin des classes lors de la construction de la représentation polonaise.

3-18) . ICDZON INTEGER (T) : tableau contenant les cardinaux des classes lors de la construction de la représentation polonaise.

3-19) . IPOL2 INTEGER (T) : tableau de travail pour la construction de la représentation polonaise.

3-20) . IPAIR1 INTEGER (T) : pour chaque paire de classes agrégeables à un niveau, tableau contenant la première classe.

3-21) . IPAIR2 INTEGER (T) : pour chaque paire de classes agrégeables à un niveau, tableau contenant la deuxième classe.

3-22) . Q REAL (T) : tableau de la matrice triangulaire inférieure des similarités.

3-23) . Q2 REAL (T) : tableau de la matrice triangulaire inférieure stricte des similarités.

3-24) . XMU REAL (T) : tableau de la contribution des représentants à la moyenne pour l'hypothèse d'absence de lien entre représentants.

3-25) . SMU REAL (T) : tableau de la contribution des représentants à la variance pour l'hypothèse d'absence de lien entre représentants.

3-26) . XNIV REAL (T) : tableau des indices des niveaux de l'arbre.

3-27) . NUMERR INTEGER (ER/ES) : code des erreurs.

4) Sous-programme(s) requis :

SIMIL - REDUC - SVSIM - NORMA - POLGL - SIGNF.

5) Sous-programme(s) appelant(s) :

CLAS2.

6) Transmission des données :

ni espaces communs, ni équivalences, ni fichiers.

1) Sous-programme POLGL :

2) Objet : construction de la représentation polonaise de l'arbre de la classification totale des représentants des classes.

3) Description des arguments :

3-1) . IFICH INTEGER (E) : tableau des numéros de fichiers.

3-2) . MXSYM INTEGER (E) : nombre maximum d'individus à classifier en même temps.

3-3) . NIPOL INTEGER (E) : dimension du tableau de la représentation polonaise.

3-4) . MIQ INTEGER (E) : dimension du tableau de la matrice triangulaire inférieure des similarités.

3-5) . MIQ2 INTEGER (E) : dimension du tableau de la matrice triangulaire inférieure stricte des similarités.

3-6) . MXPAIR INTEGER (E) : nombre maximum de paires de classes agrégeables à un même niveau de l'arbre.

3-7) . NCLTOT INTEGER (E) : nombre de représentants à classifier.

3-8) . B1 REAL (E) : voir REDUC ou STNIV.

3-9) . B2 REAL (E) : voir REDUC ou STNIV.

3-10) . B3 REAL (E) : voir REDUC ou STNIV.

3-11) . LSYM INTEGER (E) : tableau facilitant l'adressage des tables de similarités.

3-12) . Q2 REAL (E) : tableau de la matrice triangulaire inférieure stricte des similarités.

3-13) . Q REAL (ES) : tableau de la matrice triangulaire inférieure des similarités.

3-14) . NIV INTEGER (S) : plus haut niveau de l'arbre.

3-15) . IPOL INTEGER (S) : tableau de la représentation polonaise.

3-16) . XNIV REAL (S) : tableau des indices des niveaux.

3-17) . LADR INTEGER (T) : tableau permettant de repérer les classes.

3-18) . IPOL2 INTEGER (T) : tableau de travail (sauvegardes).

3-19) . IDBZON INTEGER (T) : tableau repérant le début des classes dans le tableau IPOL.

3-20) . IFNZON INTEGER (T) : tableau repérant la fin des classes dans le tableau IPOL.

3-21) . ICDZON INTEGER (T) : tableau contenant les cardinaux des classes.

3-22) . IPAIR1 INTEGER (T) : pour chaque paire de classes agrégeables à un niveau, tableau indiquant la première classe.

3-23) . IPAIR2 INTEGER (T) : pour chaque paire de classes agrégeables à un niveau, tableau indiquant la deuxième classe.

3-24) . NUMERR INTEGER (ER/ES) : code des erreurs.

4) Sous-programme(s) requis :

PRAGR - POLON - MAJR - ACTUA - STNIV.

5) Sous-programme(s) appelant(s) :

CLS22.

6) Transmission des données :

ni espaces communs, ni équivalences, ni fichiers.

7) Divers :

* mêmes techniques que POLAR.

* différences : - la classification est totale,
- l'indice de niveau est calculé pour chaque niveau en vue de déterminer les niveaux significatifs.

1) Sous-programme SIGNF :

2) Objet : déterminer les niveaux significatifs (ceux dont l'indice correspond à un maximum local).

3) Description des arguments :

3-1) . MXSYM INTEGER (E) : nombre maximum d'individus à classifier en même temps.

3-2) . NIV INTEGER (E) : plus haut niveau de l'arbre.

3-3) . XNIV REAL (E) : tableau des indices des niveaux.

3-4) . NBNVS INTEGER (S) : nombre de niveaux significatifs.

3-5) . NVSG INTEGER (S) : tableau des niveaux significatifs.

4) Sous-programme(s) requis :

aucun.

5) Sous-programme(s) appelant(s) :

CLS22.

6) Transmission des données :

ni espaces communs, ni équivalences, ni fichiers.

7) Divers :

néant.

1) Sous-programme ERROR :

2) Objet : traitement des erreurs : écriture d'un message sur le fichier de sortie.

3) Description des arguments :

3-1) . ISORT INTEGER (E) : numéro du fichier de sortie.

3-2) . NUMERR INTEGER (ER/E) : code de l'erreur.

3-3) . ICDERR INTEGER (ER/E) : dimension minimum à donner au tableau lorsque l'erreur est une dimension de tableau insuffisante.

4) Sous-programme(s) requis :

aucun.

5) Sous-programme(s) appelant(s) :

CLMIL - ALLOC - SIMIL - PRAGR - EDITO - TASS1 - TASS2.

6) Transmission des données :

- ni espaces communs, ni équivalences,
- fichier utilisé en écriture : ISORT=42.

7) Divers :

néant.

1) Sous-programme ILIRE :

2) Objet : lecture d'un tableau sur un fichier binaire.

3) Description des arguments :

3-1) . IFILE INTEGER (E) : numéro du fichier binaire.

3-2) . IDEB INTEGER (E) : indice du premier élément du tableau à lire.

3-3) . IFIN INTEGER (E) : indice du dernier élément du tableau à lire.

3-4) . IDIMS INTEGER (E) : dimension du tableau.

3-5) . ITAB INTEGER (ES) : tableau sur lequel on doit lire.

4) Sous-programme(s) requis :

aucun.

5) Sous-programme(s) appelant(s) :

LCDON - LCDNG - SIMIL - EDITO.

6) Transmission des données :

* ni espaces communs, ni équivalences,

* fichier utilisé : celui passé en paramètre - pour l'ensemble du programme les fichiers 43, 44 et 45.

7) Divers :

néant.

1) Sous-programme IECRI :

2) Objet : écriture d'un tableau sur un fichier binaire.

3) Description des arguments :

3-1) . IFILE INTEGER (E) : numéro du fichier binaire.

3-2) . IDEB INTEGER (E) : indice du premier élément du tableau à écrire.

3-3) . IFIN INTEGER (E) : indice du dernier élément du tableau à écrire.

3-4) . IDIMS INTEGER (E) : dimension du tableau.

3-5) . ITAB INTEGER (ES) : tableau que l'on veut écrire.

4) Sous-programme(s) requis :

aucun.

5) Sous-programme(s) appelant(s) :

LCDON - LCDNG - STOCK.

6) Transmission des données :

* ni espaces communs, ni équivalences,

* fichier utilisé : celui passé en paramètre - pour l'ensemble du programme les fichiers 43, 44 et 45.

7) Divers :

néant.

1) Sous-programme OUVRE :

2) Objet : simulation de l'ouverture d'un fichier (instruction REWIND).

3) Description des arguments :

3-1) . IFILE INTEGER (E) : numéro du fichier.

4) Sous-programme(s) requis :

aucun.

5) Sous-programme(s) appelant(s) :

CLMIL - CLAS1 - LCDON - LCDNG - SIMIL - EDITO.

6) Transmission des données :

- ni espaces communs, ni équivalences.
- fichier utilisé : celui passé en paramètre - En fait tous ceux utilisés par le programme : 11, 41, 42, 43, 44, 45, 46, 47, 48, 49 et 61.

7) Divers :

néant.

1) Sous-programme FERME :

2) Objet : simulation de la fermeture d'un fichier (sous-programme vide).

3) Description des arguments :

3-1) . IFILE INTEGER (E) : numéro du fichier.

4) Sous-programme(s) requis :

aucun.

5) Sous-programme(s) appelant(s) :

CIMIL - LCDON - LCDNG - SIMIL - EDITO.

6) Transmission des données :

- ni espaces communs, ni équivalences.
- fichier utilisé : celui passé en paramètre - En fait tous ceux utilisés par le programme : 11, 41, 42, 43, 44, 45, 46, 47, 48, 49 et 61.

7) Divers :

néant.

1) Sous-programme TASSO :

2) Objet : réallocation dynamique en vue de la sortie des résultats "texte" : récupération de place mémoire en éliminant les tableaux qui ne sont plus necéssaires et en dimensionnant les autres avec leur dimension exacte et non plus un majorant.

3) Description des arguments :

3-1) . LGINTG INTEGER (E) : dimension du super-tableau des entiers.

3-2) . NIND INTEGER (E) : nombre total d'individus.

3-3) . NCLTOT INTEGER (E) : nombre exact de classes.

3-4) . NBNVS INTEGER (E) : nombre exact de niveaux significatifs.

3-5) . IREP INTEGER (ES) : début du tableau des représentants.

3-6) . IOREP INTEGER (ES) : début du tableau assurant la correspondance entre les représentants et leurs classes.

3-7) . IIPOL INTEGER (ES) : début du tableau de la représentation polonaise.

3-8) . IIPOL2 INTEGER (ES) : début d'un tableau de travail.

3-9) . INVSG INTEGER (ES) : début du tableau des niveaux significatifs.

3-10) . NIPOL INTEGER (ES) : dimension du tableau de la représentation polonaise.

3-11) . INTG INTEGER (ES) : super-tableau des entiers.

3-12) . IRPOLT INTEGER (S) : début du tableau destiné à contenir les représentations polonaises des classes.

3-13) . ICDERR INTEGER (ER/S) : dimension nécessaire pour le super-tableau des entiers en cas de dépassement de capacité.

4) Sous-programme(s) requis :

aucun.

5) Sous-programme(s) appelant(s) :

appelé par le sous-programme principal CLMIL.

6) Transmission des données :

ni espaces communs, ni équivalences, ni fichiers.

7) Divers :

composition du super-tableau des entiers en sortie :

I IORDRP I INDREP I NVSG I IPOL2 I Ø I IPOLTT I

1) Sous-programme EDITO :

2) Objet : édition des résultats "texte" : niveaux significatifs, représentation polonaise de l'arbre de classification des classes et composition des classes. Si le dessin de l'arbre global a été demandé, remplacement dans la représentation polonaise du numéro actuel des classes par leur ancien numéro (avant le tri des représentants).

3) Description des arguments :

3-1) . IFICH INTEGER (E) : tableau des numéros de fichiers.

3-2) . IPARAM INTEGER (E) : tableau des paramètres.

3-3) . NCLTOT INTEGER (E) : nombre de classes.

3-4) . NIPOL INTEGER (E) : dimension du tableau de la représentation polonaise.

3-5) . NBNVS INTEGER (E) : nombre de niveaux significatifs.

3-6) . NIND2 INTEGER (E) : dimension du tableau des représentations polonaises des classes.

3-7) . INDREP INTEGER (E) : tableau des représentants.

3-8) . IORDRP INTEGER (E) : tableau assurant la correspondance entre les classes et leurs représentants.

3-9) . NVSG INTEGER (E) : tableau des niveaux significatifs.

3-10) . IPOL INTEGER (ES) : tableau de la représentation polonaise de la classification des classes.

3-11) . IPOLTT INTEGER (S) : tableau des représentations polonaises des classes.

3-12) . IPOL2 INTEGER (T) : tableau de travail.

4) Sous-programme(s) requis :

MODAR - ERROR - ILIRE - IECRI - OUVRE - FERME.

5) Sous-programme(s) appelant(s) :

appelé par le sous-programme principal CLMIL.

6) Transmission des données :

- ni espaces communs, ni équivalences,

- fichiers utilisés :

  - fichier de sortie "texte" : ISORT = 42,

  - fichier binaire en lecture des représentations polonaises des classes ITEMP1 = 45.

7) Divers :

adressage d'une classe dans le tableau des représentations polonaises des classes :

il est possible grâce au tableau IORDRP qui pour chaque classe contient le numéro de sa tranche et le numéro de la classe dans sa tranche avec le codage suivant :

IORDRP(CL)=ICARDT*(NT-1)+NUMCLAS où :
CL est la classe,
ICARDT est le cardinal d'une tranche complète,
NT est le numéro de la tranche et
NUMCLAS le numéro de la classe dans la tranche.

On se positionne directement sur le début des représentations polonaises des classes de la tranche car toute tranche complète occupe le même nombre de places dans le tableau IPOLTT (2*ICARDT - voir STOCK).
On saute ensuite les classes précédantes de la tranche (la représentation polonaise d'une tranche étant précédée de sa longueur) pour accéder à la classe désirée.

1) Sous-programme MODAR :

2) Objet : modification de la représentation polonaise d'un arbre de manière à placer à gauche les branches correspondant aux sous-classes de cardinal plus grand.

3) Description des arguments :

3-1) . IDIMS INTEGER (E) : dimension du tableau de la représentation polonaise.

3-2) . NB2 INTEGER (E) : longueur de la représentation polonaise (terminée par 0).

3-3) . IARB INTEGER (ES) : tableau de la représentation polonaise.

3-4) . IARB1 INTEGER (T) : tableau de travail (sauvegardes).

4) Sous-programme(s) requis :

UNARB.

5) Sous-programme(s) appelant(s) :

EDITO.

6) Transmission des données :

ni espaces communs, ni équivalences, ni fichiers.

7) Divers :

* on suppose qu'au plus cinquante classes s'agrègent au même niveau.

* ce sous-programme commence en appelant le sous-programme UNARB qui "uniformise" la représentation polonaise de l'arbre : lorsque plusieurs sous-classes s'agrègent au même niveau, il place en tête les numéros du niveau.

exemple :
-4 -4 a b -4 c -4 d e --> -4 -4 -4 -4 a b c d e
(les lettres représentent des sous-classes).

* algorithme :

pour chaque agrégation (en parcourant la représentation polonaise):

- détermination du nombre de sous-classes (nombre d'occurences du numéro du niveau plus un).

- tri des sous-classes par cardinal décroissant.

- réordonnement des sous-classes.

1) Sous-programme UNARB :

2) Objet : uniformise la représentation polonaise d'un arbre : met en tête les numéros des niveaux où a lieu une agrégation de plus de deux sous-classes.

3) Description des arguments :

3-1) . IDIMS INTEGER (E) : dimension du tableau de la représentation polonaise.

3-2) . NB2 INTEGER (E) : longueur de la représentation polonaise (terminée par 0).

3-3) . IARB INTEGER (ES) : tableau de la représentation polonaise.

4) Sous-programme(s) requis :

aucun.

5) Sous-programme(s) appelant(s) :

MODAR.

6) Transmission des données :

ni espaces communs, ni équivalences, ni fichiers.

7) Divers :

pour chaque agrégation (nombre négatif en parcourant la représentation polonaise):

- détermination du nombre de numéros du niveau mal placés ainsi que de l'indice de fin de la classe.
- si il existe des numéros du niveau mal placés :
  - parcours inverse de la classe en écrasant ces numéros.
  - recopie en tête de la classe de ces numéros.

1) Sous-programme TASS1 :

2) Objet : réallocation dynamique en vue du dessin de l'arbre de classification des classes, élimination de tableaux qui ne sont plus nécessaires et allocation dynamique pour les tableaux nécessaires au dessin de l'arbre.

3) Description des arguments :

3-1) . ISORT INTEGER (E) : numéro du fichier de sortie (en cas d'erreur).

3-2) . LGINTG INTEGER (E) : dimension du super-tableau des entiers.

3-3) . LGLOGI INTEGER (E) : dimension du super-tableau des booléens.

3-4) . LGICHN INTEGER (E) : dimension du super-tableau des caractères.

3-5) . NIPOL INTEGER (E) : dimension de la représentation polonaise.

3-6) . NBNVS INTEGER (E) : nombre de niveaux significatifs.

3-7) . NCLTOT INTEGER (E) : nombre de classes.

3-8) . IIPOL INTEGER (ES) : début du tableau de la représentation polonaise.

3-9) . IIPOL2 INTEGER (ES) : début d'un tableau de travail.

3-10) . INVSG INTEGER (ES) : début du tableau des niveaux significatifs.

3-11) . ICHFIN INTEGER (ES) : nombre de caractères occupés dans le super-tableau des caractères.

3-12) . INTG INTEGER (ES) : super-tableau des entiers.

3-13) . ILPHA INTEGER (S) : profondeur de la pile de travail pour dessiner l'arbre.

3-14) . IIMP INTEGER (S) : début du tableau d'une ligne du dessin de l'arbre.

3-15) . LGIMP INTEGER (S) : dimension du tableau d'une ligne du dessin de l'arbre.

3-16) . IINF INTEGER (S) : début d'un tableau de travail pour le dessin de l'arbre.

3-17) . IISUP INTEGER (S) : début d'un tableau de travail pour le dessin de l'arbre.

3-18) . INIV INTEGER (S) : début d'un tableau de travail pour le dessin de l'arbre.

3-19) . IMAX INTEGER (S) : début d'un tableau de travail pour le dessin de l'arbre.

3-20) . IMIL INTEGER (S) : début d'un tableau de travail pour le dessin de l'arbre.

3-21) . IIPILE INTEGER (S) : début d'un tableau de travail pour le dessin de l'arbre.

3-22) . IINUP INTEGER (S) : début d'un tableau de travail pour le dessin de l'arbre.

3-23) . IINU INTEGER (S) : début d'un tableau de travail pour le dessin de l'arbre.

3-24) . LBOOL INTEGER (S) : début d'un tableau de booléens pour le dessin de l'arbre.

3-25) . NUMERR INTEGER (ER/ES) : code des erreurs.

4) Sous-programme(s) requis :

ERROR.

5) Sous-programme(s) appelant(s) :

appelé par le sous-programme principal CLMIL.

6) Transmission des données :

ni espaces communs, ni équivalences, ni fichiers directement utilisés.

7) Divers :

néant.

1) Sous-programme TASS2 :

2) Objet : réallocation dynamique pour le dessin de l'arbre global et détermination de la représentation polonaise de l'arbre global.

3) Description des arguments :

3-1) . ISORT INTEGER (E) : numéro du fichier de sortie (en cas d'erreur).

3-2) . LGINTG INTEGER (E) : dimension du super-tableau des entiers.

3-3) . LGLOGI INTEGER (E) : dimension du super-tableau des booléens.

3-4) . LGICHN INTEGER (E) : dimension du super-tableau des caractères.

3-5) . NIND INTEGER (E) : nombre total d'individus.

3-6) . NIND2 INTEGER (E) : dimension du tableau de la représentation polonaise globale.

3-7) . NIPOL INTEGER (E) : dimension du tableau de la représentation polonaise de l'arbre de classification des classes.

3-8) . NCLTOT INTEGER (E) : nombre de classes.

3-9) . ICARDT INTEGER (E) : cardinal d'une tranche complète.

3-10) . IRPOLT INTEGER (E) : début du tableau des représentations polonaises des classes.

3-11) . NBNVS INTEGER (ES) : nombre de niveaux significatifs.

3-12) . INVSG INTEGER (ES) : début du tableau des niveaux significatifs.

3-13) . IIPOL INTEGER (ES) : début du tableau de la représentation polonaise de l'arbre de classification des classes.

3-14) . ICHFIN INTEGER (ES) : nombre de caractères utilisés dans le super-tableau des caractères.

3-15) . INTG INTEGER (ES) : super-tableau des entiers.

3-16) . ILPHA INTEGER (S) : profondeur de la pile de travail pour dessiner l'arbre.

3-17) . IIPLGL INTEGER (S) : début du tableau de la représentation polonaise globale.

3-18) . IIPLG2 INTEGER (S) : début d'un tableau de travail pour dessiner l'arbre global.

3-19) . IIMP INTEGER (S) : début du tableau d'une ligne du dessin de l'arbre.

3-20) . LGIMP INTEGER (S) : dimension du tableau contenant une ligne du dessin de l'arbre.

3-21) . IINF INTEGER (S) : début d'un tableau de travail pour le dessin de l'arbre.

3-22) . IISUP INTEGER (S) : début d'un tableau de travail pour le dessin de l'arbre.

3-23) . INIV INTEGER (S) : début d'un tableau de travail pour le dessin de l'arbre.

3-24) . IMAX INTEGER (S) : début d'un tableau de travail pour le dessin de l'arbre.

3-25) . IMIL INTEGER (S) : début d'un tableau de travail pour le dessin de l'arbre.

3-26) . IIPILE INTEGER (S) : début d'un tableau de travail pour le dessin de l'arbre.

3-27) . IINUP INTEGER (S) : début d'un tableau de travail pour le dessin de l'arbre.

3-28) . IINU INTEGER (S) : début d'un tableau de travail pour le dessin de l'arbre.

3-29) . LBOOL INTEGER (S) : début d'un tableau de booléens pour le dessin de l'arbre.

3-30) . ICDERR INTEGER (ER/ES) : dimension nécessaire pour le super-tableau des entiers en cas de dépassement de capacité.

3-31) . NUMERR INTEGER (ER/ES) : code des erreurs.

4) Sous-programme(s) requis :

REPGL - ERROR.

5) Sous-programme(s) appelant(s) :

appelé par le sous-programme principal CLMIL si le dessin de l'arbre global a été demandé.

6) Transmission des données :

ni espaces communs, ni équivalences, ni fichiers directement utilisés.

7) Divers :

La représentation polonaise de l'arbre global est déterminée par le sous-programme REPGL.

1) Sous-programme REPGL :

2) Objet : produire la représentation polonaise de l'arbre global à partir de la représentation polonaise de la classification des classes et de celles des classes.

3) Description des arguments :

3-1) . NIPOL INTEGER (E) : dimension du tableau de la représentation polonaise de la classification des classes.

3-2) . NIND2 INTEGER (E) : dimension des tableaux de la représentation polonaise globale et de celles de classes.

3-3) . ICARDT INTEGER (E) : cardinal d'une tranche complète.

3-4) . IPOL INTEGER (E) : tableau de la représentation polonaise de la classification des classes.

3-5) . IPOLTT INTEGER (E) : tableau des représentations polonaises des classes.

3-6) . IPOLGL INTEGER (S) : tableau de la représentation polonaise de l'arbre global.

4) Sous-programme(s) requis :

aucun.

5) Sous-programme(s) appelant(s) :

TASS2.

6) Transmission des données :

ni espaces communs, ni équivalences, ni fichiers.

7) Divers :

Pour obtenir la représentation polonaise de l'arbre global, on remplace le numéro des classes dans la représentation polonaise de la classification des classes par la représentation polonaise de ces classes. Pour l'adressage dans le tableau IPOLTT voir EDITO.

1) Sous-programme EDIT2 :

2) Objet : édition de la représentation polonaise de l'arbre global.

3) Description des arguments :

3-1) . ISORT INTEGER (E) : numéro du fichier de sortie.

3-2) . NIND2 INTEGER (E) : dimension du tableau de la représentation polonaise de l'arbre global.

3-3) . IPOLGL INTEGER (E) : tableau de la représentation polonaise de l'arbre global.

4) Sous-programme(s) requis :

aucun.

5) Sous-programme(s) appelant(s) :

appelé par le sous-programme principal CLMIL.

6) Transmission des données :

- ni espaces communs, ni équivalences.
- fichier utilisé en sortie : ISORT = 42.

7) Divers :

néant.

1) Sous-programme ARBRE :

2) Objet : détermination de la représentation polonaise de l'arbre condensé à ses niveaux significatifs puis dessin de cet arbre.

3) Description des arguments :

3-1) . ISOR INTEGER (E) : numéro du fichier où sera imprimé le début du dessin de l'arbre.

3-2) . IFIPAS INTEGER (E) : "pas" entre les différents fichiers où sera dessiné l'arbre.

3-3) . ITITRE INTEGER (E) : tableau contenant le titre de la classification.

3-4) . IFMTID INTEGER (E) : tableau pouvant contenir le format de sortie d'une ligne de l'arbre.

3-5) . NB INTEGER (E) : nombre d'objets présents dans l'arbre.

3-6) . NBB INTEGER (E) : dimension du tableau de la représentation polonaise.

3-7) . NBNVS INTEGER (E) : nombre de niveaux significatifs.

3-8) . NVP1 INTEGER (E) : nombre de niveaux de l'arbre condensé.

3-9) . ILPHA INTEGER (E) : profondeur de la pile.

3-10) . LGIMP INTEGER (E) : dimension du tableau du dessin d'une ligne de l'arbre.

3-11) . NITEM INTEGER (E) : nombre d'objets dans le cas où l'utilisateur a fourni leurs noms, un sinon.

3-12) . ICAR INTEGER (E) : nombre maximum de mots mémoire nécessaires pour stocker le nom d'un objet.

3-13) . IOPTIO INTEGER (E) : option de sortie.

3-14) . NIVSIG INTEGER (E) : tableau des niveaux significatifs.

3-15) . ITEM INTEGER (E) : tableau bidimensionnel contenant lorsqu'ils ont été fournis les noms des individus.

3-16) . IPOL INTEGER (ES) : tableau de la représentation polonaise (en entrée) puis pour les calculs tableau de la représentation polonaise de l'arbre condensé.

3-17) . IMP INTEGER (T) : tableau contenant le dessin d'une ligne de l'arbre.

3-18) . INF INTEGER (T) : pile contenant le numéro de la ligne inférieure des sous-classes agrégées à un niveau.

3-19) . ISUP INTEGER (T) : pile contenant le numéro de la ligne supérieure des sous-classes agrégées à un niveau.

3-20) . NIV INTEGER (T) : pile contenant le niveau où se connectent des sous-classes (connectée avec INF et ISUP).

3-21) . MAX INTEGER (T) : tableau indiquant le nombre de places utilisées à chaque niveau dans les tableaux bidimensionnels IPILE et INUP.

3-22) . IPOL2 INTEGER (T) tableau contenant les numéros réels des niveaux d'agrégation (les niveaux significatifs sont positifs) double le tableau IPOL.

3-23) . MIL INTEGER (T) : tableau indiquant pour chaque niveau dans le cas où une agrégation est en cours à ce niveau le numéro de la ligne où doit apparaître un trait horizontal au niveau suivant (milieu de la classe).

3-24) . IPILE INTEGER (T) : tableau bidimensionnel contenant pour chaque niveau (condensé) les numéros des lignes où l'on doit inscrire les numéros des niveaux réels d'agrégation.

3-25) . INU INTEGER (T) : tableau bidimensionnel des numéros des niveaux réels (connecté au tableau IPILE).

3-26) . NU INTEGER (T) : tableau contenant les numéros des objets présents dans le dessin de l'arbre (double IPOL).

3-27) . BOOL LOGICAL (T) : tableau indiquant pour chaque niveau de l'arbre condensé si une verticale signifiant l'agrégation d'une classe a été initialisée.

4) Sous-programme(s) requis :

aucun.

5) Sous-programme(s) appelant(s) :

appelé par le sous-programme principal CLMIL.

6) Transmission des données :

- ni espaces communs, ni équivalences,
- fichiers utilisés : fichiers de sortie "dessin" ISOR = 46 et éventuellement 47, 48 et 49.

7) Divers :

* condensation de l'arbre :

les niveaux retenus pour la condensation de l'arbre sont les niveaux significatifs et le dernier niveau, les autres niveaux sont réhaussés au niveau retenu directement supérieur.

Composition de quelques tableaux après la condensation :

IPOL : la représentation polonaise condensée où les numéros des n individus ont été remplacés par les n premiers entiers impairs (il s'agit en fait des numéros des lignes où apparaîtront les terminaux représentants ces individus dans le dessin).

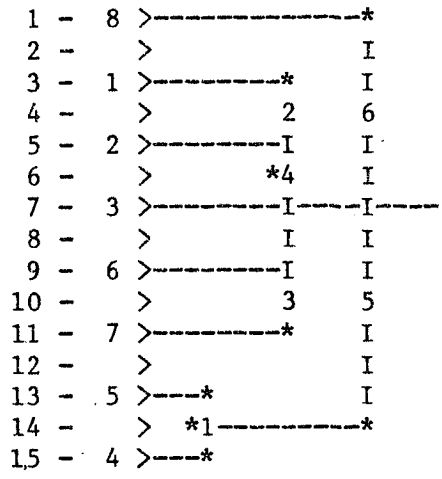
IPOL2 : les numéros réels des niveaux (non condensés) à la même adresse où ils étaient dans IPOL avant la condensation (les niveaux significatifs sont positifs).

NU : les numéros des objets.

exemple : représentation polonaise non condensée à la première ligne, niveaux significatifs 1 et 4.

	-6	8	-5	-4	-2	1	2	-3	-3	3	6	7	-1	5	4	0
IPOL	-3	1	-3	-2	-2	3	5	-2	-2	7	9	11	-1	13	15	0
IPOL2	-6	0	-5	4	-2	0	0	-3	-3	0	0	0	1	0	0	0
NU	8	1	2	3	6	7	5	4								

dessin de l'arbre :



* dessin d'une ligne L :

Pour dessiner l'arbre, le tableau IPOL est parcouru feuille par feuille, ce qui fait dessiner l'arbre ligne par ligne.

Principaux tableaux et variables utilisés :

IPILE (*,N) : pour chaque niveau N : suite des numéros de ligne où doivent être placés les numéros des niveaux réels, IPILE(1,N) contient le numéro de la ligne où s'achève l'agrégation éventuelle en cours au niveau (condensé) N.

INU (*,*) : double le tableau IPILE et contient les valeurs réelles respectives des niveaux.

MAX (N) : pour chaque niveau N indique le nombre de cases remplies dans les tableaux bidimensionnels IPILE et INU pour ce niveau.

BOOL (N) : pour chaque niveau N, indique si une verticale signifiant l'agrégation de sous-classes a été initialisée à une ligne précédente.

MIL (N) : pour chaque niveau N, indique le numéro de la ligne où doit apparaître un trait horizontal commençant une branche pour la nouvelle classe qui s'est formée à ce niveau.

BASC : booléen indiquant si l'on doit tracer un trait plein sur la ligne courante au niveau courant.

Sur l'exemple précédant après traitement de IPOL(2) :

IPILE(1,3)=15 (ligne de fermeture de la classe),
IPILE(2,3)=4, INU(2,3)=6,
IPILE(3,3)=10, INU(3,3)=5,
MAX(3)=3,
BOOL(3)=.TRUE.,
MIL(3)=7.

Une ligne est dessinée du niveau 0 (feuille) au niveau 1, du niveau 1 au niveau 2 et ainsi de suite. Nous appellerons le dessin consécutif au passage d'un niveau au suivant un segment de ligne. Chaque segment est composé de deux parties :

Première partie :

soit le prolongement d'une branche si BASC=.TRUE.,
soit rien (des blancs) sinon.

Initialisation de BASC : mis à .TRUE. au début de chaque ligne impaire (feuille de l'arbre) et à .FALSE. au début de chaque ligne paire.

mise à jour : à .TRUE. quand pour le niveau courant K, MIL(K)=L (début d'une branche).

à .FALSE. quand pour le niveau courant K, on initialise une ligne verticale d'agrégation, ou alors si une telle verticale a été initialisée auparavant (BOOL(K)=.TRUE.)

Deuxième partie :

1) pour cette ligne et pour ce niveau K une verticale d'agrégation a été initialisée BOOL(K)=.TRUE. :

cas 1) : IPILE(1,K)=L : on est sur la ligne de fermeture de la classe, on imprime '*',

cas 2) : le numéro de la ligne courante est dans IPILE pour ce niveau (IPILE(x,K)=L) : on imprime INU(x,K) numéro du niveau de l'arbre non condens'où s'effectue l'agrégation (il peut être significatif INU(x,K)<0).

cas 3) : autre on imprime 'I' (trait vertical).

2) on doit initialiser une verticale : on imprime '*' et on initialise les tableaux (voir plus loin).

3) autre : on imprime ce qu'on a imprimé avant ('-' ou ' ').

* initialisation d'une verticale au niveau K :

quand :

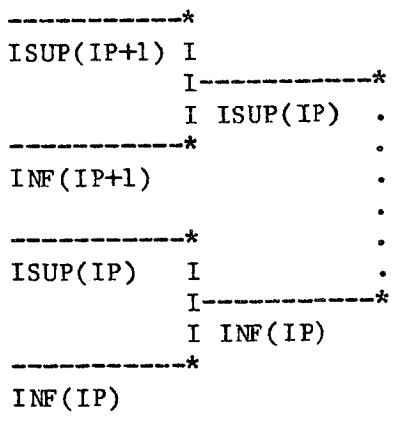
il s'agit du début de la formation d'une classe, on l'initialise quand on rencontre dans IPOL le premier individu de cette classe c'est à dire :
quand l'élément de IPOL sur lequel on pointe appartient à une classe agrégée au niveau courant K et que la verticale correspondante n'a pas déjà été initialisée.

que doit-on faire ?

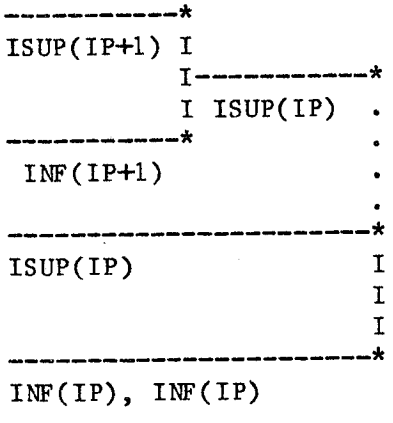
mettre BOOL(K) à .TRUE., BASC à .FALSE. et initialiser les tableaux IPILE(*,K), INU(*,K), MAX(K) et MIL(K).

remplissage des tableaux :

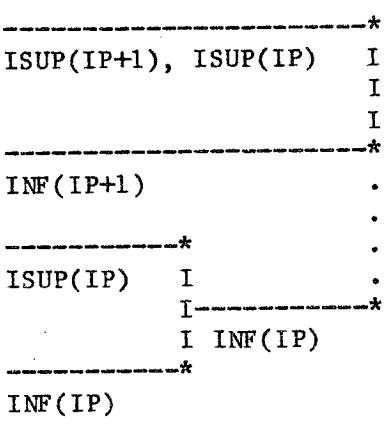
Pour ce faire on va parcourir de droite à gauche la représentation polonaise de la classe s'agrégeant à ce niveau. Il s'agit de repérer les lignes où seront inscrits sur la verticale d'agrégation les numéros des noeuds correspondant aux différentes sous-classes composantes. Les numéros de ces lignes seront enregistrés dans le tableau IPILE. Pour repérer ces numéros, on va se servir de trois piles connectées INF, ISUP et NIV qui contiennent respectivement la ligne inférieure, la ligne supérieure et le niveau de formation de la sous-classe déjà rencontrée en parcourant de droite à gauche la classe en question. La structure des sous-classes composantes peut prendre les quatre formes suivantes notées A, B, C et D; IP est le pointeur des sommets de piles.



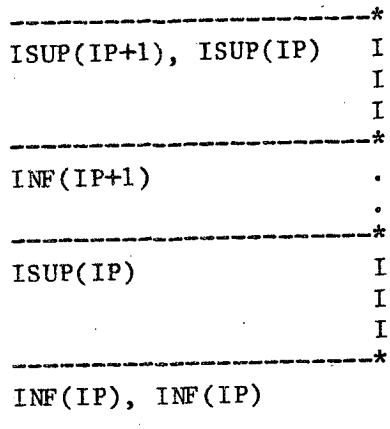
CAS A



CAS B



CAS C



CAS D

Soit S la valeur absolue du dernier niveau rencontré lors du parcours inverse de la représentation polonaise de la classe, les cas A, B, C et D sont respectivement caractérisés par :

A : NIV(IP) ≠ S et NIV(IP+1) ≠ S
B : NIV(IP) = S et NIV(IP+1) ≠ S
C : NIV(IP) ≠ S et NIV(IP+1) = S
D : NIV(IP) = S et NIV(IP+1) = S

Chargement des piles INF, ISUP et NIV :

On note IP le pointeur des sommets de piles (initialement IP=0) et on parcourt la représentation polonaise de la classe de droite á gauche.

* à la rencontre d'une feuille : empilement
  IP=IP+1; INF(IP)=ISUP(IP)=numéro de la feuille; NIV(IP)=0;

* à la rencontre d'un niveau (entier négatif -S) dépilement :

  IP=IP-1 puis aiguillage suivant le type de la situation (A, B, C ou D)

  A : INF(IP)=1/2(INF(IP)+ISUP(IP));
      ISUP(IP)=1/2(INF(IP+1)+ISUP(IP+1))

  B : INF(IP)=INF(IP); ISUP(IP)=1/2(INF(IP+1)+ISUP(IP+1))

  C : INF(IP)=1/2(INF(IP)+ISUP(IP)); ISUP(IP)=ISUP(IP+1)

  D : INF(IP)=INF(IP); ISUP(IP)=ISUP(IP+1)

Lorsqu'on tombe sur un niveau S=K, on charge IPILE(MAX(K),K) avec 1/2(INF(IP)+ISUP(IP)) et INU(MAX(K),K) avec le niveau réel contenu dans IPOL2 sans oublier d'incrémenter MAX(K) de un. On détermine au fur et à mesure la valeur de MIL(K); IPILE(1,K)

contiendra le numéro de la ligne d'apparition sur le dessin de la dernière feuille de la classe. L'algorithme se poursuit ainsi jusqu'à ce que l'on ait parcouru toute la représentation polonaise de cette classe.

Imprimé en France

par

l'Institut National de Recherche en Informatique et en Automatique